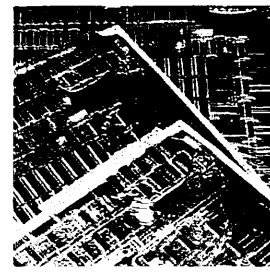
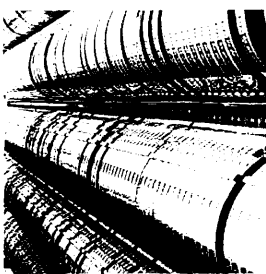


Prime Computer, Inc.

**IDR6107  
EMACS Primer  
Revision 18.3**



# **EMACS Primer**

**IDR6107**

**Revision 0**

**by**

**Barbara Buck**

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 18 (Rev. 18).

**Prime Computer, Inc.  
500 Old Connecticut Path  
Framingham, Massachusetts 01701**

## COPYRIGHT INFORMATION

The information in this document is subject to change without notice and should not be construed as a commitment by Prime Computer Corporation. Prime Computer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Copyright © 1982 by  
Prime Computer, Incorporated  
500 Old Connecticut Path  
Framingham, Massachusetts 01701

PRIME and PRIMOS are registered trademarks of Prime Computer, Inc.

PRIMENET, RINGNET, and THE PROGRAMMER'S COMPANION are trademarks of Prime Computer, Inc.

## HOW TO ORDER TECHNICAL DOCUMENTS

### U.S. Customers

Software Distribution  
Prime Computer, Inc.  
1 New York Ave.  
Framingham, MA 01701  
(617) 879-2960 X2053, 2054

### Customers Outside U.S.

Contact your local Prime  
subsidiary or distributor.

### Prime Employees

Communications Services  
MS 15-13, Prime Park  
Natick, MA 01760  
(617) 655-8000, X4837

### INFORMATION Systems

Contact your Prime  
INFORMATION system dealer.

PRINTING HISTORY — EMACS PRIMER

<u>Edition</u>	<u>Date</u>	<u>Number</u>	<u>Documents Rev.</u>
Rev. 0	April, 1982	IDR6107	18.3

SUGGESTION BOX

All correspondence on suggested changes to this document should be directed to:

Barbara Buck  
Technical Publications Department  
Prime Computer, Inc.  
500 Old Connecticut Path  
Framingham, Massachusetts 01701



# Contents

About This Book	ix
1 INTRODUCTION	
Organization	1-1
2 EMACS PRELIMINARIES	
EMACS Command Conventions	2-1
Escape Commands	2-2
Control Commands	2-2
EMACS Typing Conventions	2-3
Beginning and Ending a Session	2-4
3 EMACS BASICS	
Starting EMACS	3-1
The LOAD Command	3-3
Creating a File	3-3
Entering Text	3-4
Moving the Cursor	3-5
Command Review	3-24
Making Corrections to Text	3-25
Deleting Characters	3-25
Deleting Words	3-31
Inserting Characters and Words	3-39
Saving a Document	3-47
Exiting EMACS	3-48
Summary of Basic EMACS Commands	3-50
4 MORE EMACS	
Restarting EMACS	4-1
Adding Text and Moving the Cursor	4-3
Moving the Cursor to the End of a File	4-4
Adding Text to the End of a File	4-4
Moving the Cursor to the Beginning of a File	4-5
Moving the Cursor Forward One Screen	4-6
Moving the Cursor Backward One Screen	4-7

	Adding Text to the Middle and Beginning of a File	4-12
	Command Summary	4-19
5	MOVING TEXT	
	Deleting and Yanking Lines of Text	5-1
	Deleting and Yanking Regions of Text	5-12
	Moving Text	5-18
	Erasing Incorrectly Yanked Text	5-20
	Command Summary	5-21
6	HELP AND HELPFUL HINTS	
	Aborting Running EMACS Commands	6-1
	Aborting Partially Invoked EMACS Commands	6-2
	Refreshing the Screen	6-2
	Help Commands	6-3
	Remembering Commands	6-8
	Responding to Specific Situations	6-8
	Command Summary	6-9
7	REPLACEMENT COMMANDS	
	Simple Replacement	7-2
	Query Replacement	7-4
	Command Summary	7-9
8	FORMATTING COMMANDS	
	Indenting Lines of Text	8-1
	Centering a Line	8-3
	Using Tabs	8-4
	Default Tab Stops	8-5
	Modifying the Default Interval	8-9
	Setting Your Own Tab Stops	8-11
	Setting Tabs From a Table	8-12
	Moving Text to Tab Stops	8-15
	Filling Paragraphs	8-16
	Command Summary	8-21
9	HELPFUL EDITING COMMANDS	
	Moving the Cursor Backward One Sentence	9-1
	Moving the Cursor Backward One Paragraph	9-3
	Fixing Transposed Characters	9-5
	Fixing Transposed Words	9-8
	Converting Words to Uppercase	9-10
	Converting Words to Lowercase	9-13

	Command Summary	9-15
10	OTHER USEFUL COMMANDS	
	Moving the Screen Horizontally	10-1
	Overlay Mode	10-4
	Executing PRIMOS Commands	10-5
	Command Summary	10-7
11	BUFFERS AND WINDOWS	
	Buffers	11-1
	Creating Buffers	11-1
	Using Multiple Buffers	11-3
	Getting a List of Buffers	11-7
	Saving Text in Buffers	11-9
	Multiple Windows	11-12
	Command Summary	11-19
12	KEYBOARD MACROS	
	Defining a Keyboard Macro	12-1
	Executing a Macro	12-4
	Defining a Second Macro	12-6
	Command Summary	12-15
APPENDIXES		
A	BASIC COMMAND SUMMARY	A-1
B	COMPLETE ALPHABETICAL SUMMARY OF COMMANDS	B-1
C	COMPLETE CROSS-REFERENCE LIST OF COMMANDS	C-1
	INDEX	X-1

# About This Book

This book explains how to use the EMACS editor in a tutorial fashion. It tells you exactly what you need to know to begin using EMACS immediately.

You do not need to know anything about programming or screen editors to use this book. In fact, you need only a basic understanding of PRIMOS and its file system to begin working with EMACS.

This book is divided into twelve chapters. Each chapter provides step-by-step instructions for you to follow. The instructions are accompanied by diagrams indicating what the terminal screen should look like after you've typed different commands.

## HOW TO USE THIS BOOK

The way you use this book will depend on your level of experience and what you want to accomplish with EMACS. This book will take you through the steps necessary to learn basic EMACS commands. After completing it, you will know enough EMACS commands to accomplish most editing tasks.

If you have no programming experience and little or no experience with screen editors, you will probably want to work through the entire book. After you become more familiar with EMACS, you can look through the EMACS Reference Guide (IDR5026) to find out about other things you can do with EMACS.

If you are already familiar with EMACS or if you are an experienced user of screen editors, you may want to read chapters 2, 3, and 6, and then move on to the EMACS Reference Guide. An alternative route is to proceed directly to the TEACH-EMACS files after reading Chapter 2.

#### OTHER EMACS DOCUMENTATION

You can use the commands described in this book on any terminal containing ESCAPE and CONTROL keys. If you are working on a Prime PT45 terminal, you can use its function keys in place of many commands described in this book. The PT45 EMACS Primer, (IDR5734), explains how to do this.

This book presents only a subset of EMACS commands. If you want to learn about other EMACS commands, or if you want more information about the commands you've learned, you should consult the EMACS Reference Guide. It contains a complete description of all EMACS commands grouped by function.

If you want to learn how to use the EMACS extension language to write your own functions and take advantage of EMACS' programming capabilities, see the EMACS Extension Writing Guide (IDR5025).

In addition to the printed documentation discussed above, TEACH-EMACS, an on-line tutorial, is available. It consists of three files, each describing a different subset of basic EMACS commands. TEACH-EMACS is self-instructive and quickly paced. It allows the experienced user to learn a large subset of EMACS commands in a very short time. You can find TEACH-EMACS in the directory EMACS\*.

#### A WORD ABOUT EMACS

EMACS is a full-screen display editor. This means that instead of working only with lines in your file, you can work with characters or large blocks of text as well and see the results of your changes immediately.

When you use EMACS for the first time, it may seem a bit overwhelming because it uses many commands and works differently from other editors you may be accustomed to. Don't let this throw you. It is very normal to feel a little uncomfortable at first. After a short time, you will become accustomed to the way EMACS works and using EMACS commands will become second nature.

Keep in mind that EMACS has unlimited capabilities. Once you become familiar with a basic set of commands, you should consult other documentation to find out about other things EMACS can do to make your work easier.

## ACKNOWLEDGEMENTS

Prime's version of the EMACS editor was written by Software Arts, Inc. of Cambridge, Massachusetts. The user interface and the extensibility were modelled after the EMACS editor at MIT.

The EMACS concept of a screen-based, user-extensible editor was evolved over years by researchers at the MIT Laboratory for Computer Science and the MIT Artificial Intelligence Laboratory. It has been implemented on a number of different computers at MIT.

The first six chapters of this book are based on the work of Lilli Filichia. The author wishes to thank Barry Kingsbury and all others who reviewed and contributed to this document.

## CHAPTER 1

## INTRODUCTION

This book is a step-by-step tutorial that will teach you how to use the EMACS editor. You don't have to know anything about programming or screen editors to use this book. All you need is a basic understanding of PRIMOS and its file system. Everything else is explained in the following pages.

This book presents a basic set of EMACS commands only. If you need more detailed information, you should consult the EMACS Reference Guide. It contains a complete description of all EMACS commands.

## ORGANIZATION

This book contains twelve chapters. Chapters 1-6 explain commands and information that are essential for using EMACS. Chapters 7-12 introduce additional commands that will make your editing task easier. Each chapter contains step-by-step instructions, accompanied by illustrations showing you what your screen should look like after you use various commands.

Examples in each chapter contain commands learned in previous chapters. Therefore, do not move on to a new chapter until you feel confident using the commands you've already learned. A command summary follows each chapter to help you remember the new commands you've learned.

The chapters are organized as follows:

Chapter 1 is an introduction to the book. You are reading it right now.

Chapter 2 explains the EMACS commands and typing conventions used throughout this book. You should be familiar with the contents of this chapter before moving on to other chapters.

Chapter 3 introduces a basic set of EMACS commands. It explains how to start and exit EMACS, how to create and save a file, how to enter text and make simple corrections to it, and how to move around in a file.

Chapter 4 adds to the basic set of commands introduced in Chapter 3. It contains more commands for moving through a file, including search commands.

Chapter 5 explains how to move and delete text in a file.

Chapter 6 is a help chapter. It explains how to use the EMACS help commands and provides some answers for problems common to beginners. If you encounter a problem that you can't get out of while you're working with the tutorial, you can refer to Chapter 6 for help.

Chapter 7 explains how to use the commands for replacing text.

Chapter 8 describes a group of commands that are useful for formatting documents.

Chapters 9 and 10 introduce some commands that are helpful for doing things like creating and editing tables and writing reports.

Chapter 11 explains EMACS' file and buffer system and explains how to use multiple windows.

Chapter 12 explains how to create and use keyboard macros.

This book also contains three appendixes containing command summaries for easy reference. Appendix A lists the commands discussed in this book and the chapters where they are introduced. Appendix B summarizes all EMACS commands alphabetically. Appendix C groups all EMACS commands by function.



## CHAPTER 2

## EMACS PRELIMINARIES

EMACS is a screen editor. You will be using EMACS to create and edit documents. You communicate with EMACS by depressing keys (typing) on your terminal keyboard. EMACS treats certain combinations of characters as EMACS commands. EMACS treats all other characters as text that it adds to your document. Let's see how EMACS distinguishes commands from ordinary text.

## EMACS COMMAND CONVENTIONS

You can use EMACS with a variety of terminals. A typical terminal keyboard looks like this:

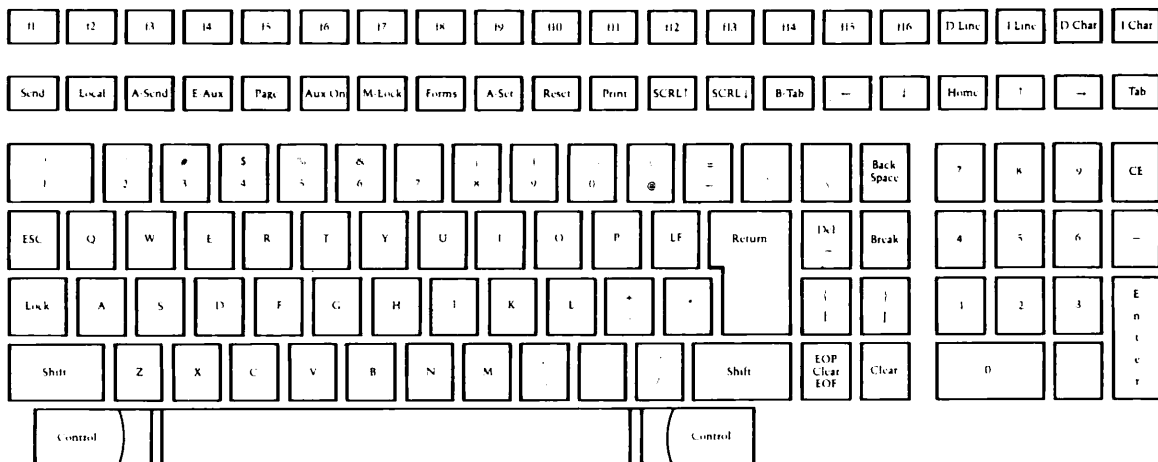


Figure 2-1. A Typical Terminal Keyboard

In the picture of the typical keyboard above, several keys are shaded in. One shaded key is the ESC key; another shaded key is the CONTROL key. These two keys have a special meaning to EMACS.

All EMACS commands begin with either the ESC key or the CONTROL key. Therefore, whenever you depress the ESC key and another character or the CONTROL key and another character, EMACS knows you're giving it a command.

### ESCAPE Commands

In this manual, EMACS commands that begin with the ESC key are written:

ESCAPE <character>

For instance, the first ESCAPE command you learn in Chapter 3 is the ESCAPE B command. When you are asked to type:

{ESC}B

it means you depress the ESC key once and let go, then you depress the B key once and let go. If you are asked to type:

{ESC}F

three times, it means you depress the ESC key once and let go, then you depress the F key once and let go; depress the ESC key once again and let go, then depress the F key once again and let go; depress the ESC key one more time and let go, then depress the F key one more time and let go. If you are asked to type:

{ESC}5{ESC}B

this means you depress the ESC key once and let it go, depress the 5 key and let it go, depress the ESC key once again and let it go, then depress the B key.

When you use the ESC key, do not hold it down longer than it takes to depress it and release it. If you do hold the ESC key down too long, EMACS misinterprets the command you are trying to type, and the message:

PL:

appears at the bottom of your screen in front of the cursor. If this happens to you, just hit the RETURN key until it goes away, or see Chapter 6 for help.

### CONTROL Commands

Commands that begin with the CONTROL key are written:

CONTROL-<character>

The first CONTROL command you learn in Chapter 3 is the CONTROL-B command. Type:

{CNTRL-B}

This means: you depress the CONTROL key, and hold it down while you depress the B key. If you are asked to type:

```
{CNTRL-F}
```

three times, you depress the CONTROL key and hold it down while you depress the F key three times. If you are asked to type:

```
{CNTRL-X}{CNTRL-W}
```

this means you depress the CONTROL key and hold it down while you first depress the X key and let it go. Then, still depressing the CONTROL key, you depress the W key.

Notice the difference between typing an ESCAPE command and a CONTROL command: you do not hold down the ESC key when you type the command character; you do hold down the CONTROL key when you type the command character.

### Combinations of Commands

Some EMACS commands are combinations of ESCAPE and CONTROL commands. If you are asked to type:

```
{ESC}5{CNTRL-F}
```

it means you depress the ESC key once and let go, you depress the 5 key once and let go, then you depress the CONTROL key and hold it down while you depress the F key.

If you are asked to type:

```
{ESC}{CNTRL-Y}
```

it means you depress the ESC key once and let it go, then you depress the CONTROL key and hold it down while you depress the Y key.

### EMACS TYPING CONVENTIONS

EMACS treats all characters that are not prefixed by an ESC key or a CONTROL key as text that it adds to your document. When you use EMACS to enter a new document, simply treat your terminal keyboard as if it were an electric typewriter: depress the SPACE key to enter a blank space; depress the RETURN key at the end of each line.

Use of the RETURN key is a little tricky in EMACS. Therefore, when you're working on the examples in this manual, only depress the RETURN key when you see the characters:

```
{RETURN}
```

### Correcting Errors

If you make a typing mistake while you are entering text into a document, you can use the BACKSPACE key to erase characters. Stop the cursor when you have erased the erroneous character and retype the rest of the line correctly. Later chapters of this manual describe other methods for correcting typing errors.

### BEGINNING AND ENDING A SESSION

A terminal session is a period of time spent at the terminal communicating with the computer. A terminal session begins when you issue the LOGIN command and ends when you issue the LOGOUT command. Chapter 2 of the New User's Guide to EDITOR and RUNOFF, explains both of these commands.

An EMACS session begins when you start EMACS and ends when you exit EMACS. EMACS sessions take place during terminal sessions. You can start and exit EMACS as many times as you want during a terminal session.

As long as you work through the chapters in order, you can work on as many chapters of this manual as you wish per EMACS session. This means you can start EMACS, work through one or more chapters, exit EMACS, and at some later time start EMACS again and work through the next chapter.

When you are ready to exit EMACS, you must first save your document in a file to make your editing changes permanent. Chapter 3 describes the procedures for saving a file and exiting EMACS. If you are unfamiliar with Prime's file system and file-naming conventions, see Chapter 2 of the New User's Guide to EDITOR and RUNOFF or Chapter 2 of the PRIME User's Guide, for help.

Each chapter in this book bases its examples on the document you create in Chapter 3. From Chapter 4 on, whenever you start EMACS to begin a new editing session, you will have to tell EMACS to find the document on which you want to work. Chapter 4 describes this procedure.

Each chapter begins where the previous one ends. Therefore, if you are trying the examples, you must complete all the examples in a chapter before you move on to the next chapter. If you fail to complete a chapter before moving on, your document will not correspond to the examples in later chapters.

## CHAPTER 3

## EMACS BASICS

This chapter demonstrates a basic set of EMACS commands. If you never learn more about EMACS than these basics, you will still find EMACS a useful and efficient editor.

The easiest way to learn the basic EMACS commands is to use them. In this chapter, at least one example follows the explanation of each command. Try the examples as you read about each command. When you finish the examples in this chapter, you will know how to:

- Start EMACS
- Create and name a new file
- Enter a document
- Move the cursor backward, forward, up, and down
- Make corrections to text
- Save a file
- Exit EMACS

If you need to exit EMACS before you complete this chapter, see EXITING EMACS at the end of the chapter for instructions. If you make a mistake and you don't know how to correct it, see EXITING EMACS at the end of this chapter, or refer to Chapter 6 for help.

## STARTING EMACS

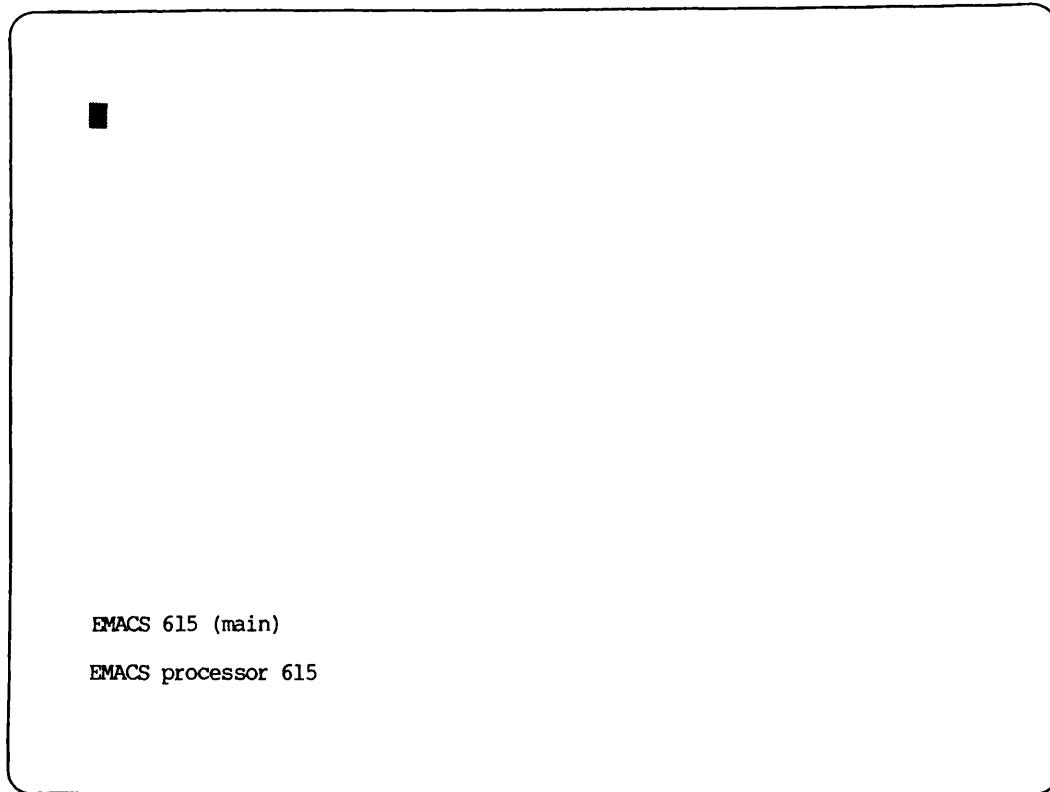
Turn on your terminal and log in. Now start EMACS. Type the following command line, substituting the name of the terminal you are currently working on for <terminal type>:

```
EMACS -TTP <terminal type> {RETURN}
```

For example, if you are working on a Prime PT45, you should type:

```
EMACS -TTP PT45 {RETURN}
```

A few seconds after you type these EMACS start-up commands, your screen will look like this:



The rectangular box in the upper left-hand corner is the cursor. The line of information near the bottom of your screen is called the mode line. You will learn more about the mode line as you work through the examples in this book.

### The LOAD Command

Before you begin working with EMACS, you should always type the ESCAPE X LOAD command. This command tells EMACS to make some special command library files available to you. If you forget to type this command, certain EMACS will not work. Try it now. Type:

```
{ESC}X LOAD {RETURN}
```

You will see some notations appear under the mode line. These notations are there to tell you that EMACS is LOADING the library files as you requested. In a few seconds, you will see the notation:

```
Autoload complete
```

At this point, EMACS is ready and waiting for you to tell it what to do.

### CREATING A FILE

When you finish entering and editing a memo, letter, or program, you'll want to save your document in a file so you can access it in the future. The CONTROL-X CONTROL-F command creates and names the file in which EMACS can later save your document. Type:

```
{CNTRL-X}{CNTRL-F}
```

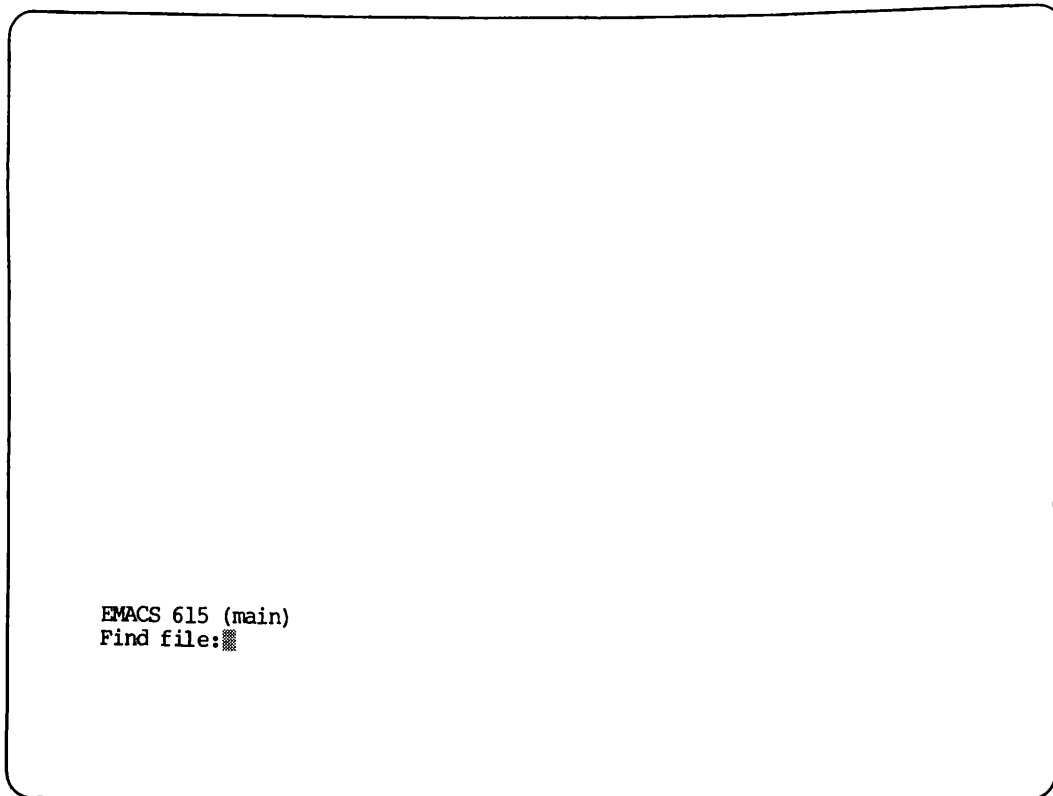
If you need help, you can refer to the section entitled EMACS COMMAND CONVENTIONS in Chapter 2.

EMACS responds to the CONTROL-X CONTROL-F command by printing the words:

```
Find file:
```

on your screen below the mode line.

Your screen should look like this:



EMACS wants you to type in a name for the file it is about to create. Let's name this file "practice". Type:

```
practice{RETURN}
```

In a few seconds, EMACS responds with a message similar to this one on the last line of your screen:

```
Opening: <MISCEL>LILLI>PRACTICE: Not found.
```

This message indicates that EMACS has not found an existing file named "practice" and therefore has created a new one for you. This new file is currently empty. The document you are about to enter will be saved in it.

#### ENTERING TEXT

You now need a document on which to practice a few EMACS commands. Let's use EMACS to enter a practice paragraph. To do this, just type the following text. (Refer to EMACS TYPING CONVENTIONS in Chapter 2, if you need help.) Remember that you can use the BACKSPACE key to make corrections at any time.



Type:

```
A farmer's daughter had been out to milk{RETURN}
the cows, and was returning to the dairy{RETURN}
carrying her pail of milk upon her head.{RETURN}
As she walked along, she began to daydream{RETURN}
in the following manner:{RETURN}
"The milk in this pail will provide me with cream,{RETURN}
which I will make into butter and take to market{RETURN}
to sell. With the money I will buy a number of eggs,{RETURN}
and these, when hatched, will produce chickens,{RETURN}
and by and by I shall have quite a large poultry-yard.
```

Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Don't worry if you failed to correct all your typing errors when they occurred. You'll be able to go back and correct them after you've read the next two chapters.

### MOVING THE CURSOR

In EMACS, the cursor marks the spot where text is entered into your document. As you typed the practice paragraph, you probably noticed that the cursor moves to the right when you strike a character. It moves to the left-hand margin of the next line when you strike the RETURN key.

The cursor marks the spot at which EMACS commands take effect. Some EMACS commands affect text to the left of the cursor, some affect text to the right of the cursor, and some affect only the character over which the cursor is positioned.

The most basic EMACS cursor commands move the cursor to a new position that is either backward, forward, up, or down from the current cursor position.

### Moving the Cursor Backward One Character

The CONTROL-B command moves the cursor backward one character. Try it. Type:

{CNTRL-B}

and watch the cursor move backward one character:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard."

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Next, type:

{CNTRL-B}

three times and watch the cursor move backward three characters:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard."

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now type:

{CNTRL-B}

enough times to move the cursor backward to the left hand margin and up over the last character of the previous line. Your screen should look like this:

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:  
"The milk in this pail will provide me with cream,  
which I will make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens  
and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

There is a shortcut to issuing the same command more than once. This shortcut is the ESCAPE <digit> command. The ESCAPE <digit> command tells EMACS how many times you want a command repeated. So, when you use the ESCAPE <digit> command, you choose the digit and then substitute it for the <digit> notation.

Combine an ESCAPE <digit> command with the CONTROL-B command and watch what happens. To move the cursor backward 10 spaces to the "e" in the word "produce," type:

```
{ESC}10{CNTRL-B}
```

Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Moving the Cursor Backward One Word

Sometimes, when editing a document, it will be more efficient for you to back space one word at a time rather than one character at a time. The ESCAPE B command moves the cursor backward one word. Type:

{ESC}B

and notice the new cursor position:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

The cursor moves from the last character in the word "produce" to the first character in the word. The first time the ESCAPE B command is issued, it moves the cursor backward to the beginning of the current word.

Now, type:

{ESC}B

once again and watch the cursor move backward to the beginning of the previous word:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now type:

{ESC}B

enough times to move the cursor backward to the beginning of the line, and then up to the first character of the last word in the previous line. Your screen should look like this:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of @ggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE



Again, you can use the shortcut if you want to type this command more than once. Try combining the ESCAPE <digit> command with the ESCAPE B command. Move the cursor to the beginning of the word "will". Type:

```
{ESC}5{ESC}B
```

Your screen should look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Moving the Cursor Forward One Character

The CONTROL-F command moves the cursor forward character by character.  
Type:

{CNTRL-F}

four times and watch the cursor move four characters to the right:

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:  
"The milk in this pail will provide me with cream,  
which I will make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Combine an ESCAPE <digit> command with CONTROL-F to move the cursor forward 16 more spaces. Type:

```
{ESC}16 {CNTRL-F}
```

so that your screen looks like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Next, move the cursor down to the first character of the next line. You can do this by typing:

```
{CNTRL-F}
```

seven times or by typing:

```
{ESC}7{CNTRL-F}
```

Your screen should now look like this:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Moving the Cursor Forward One Word

It will sometimes be more efficient for you to move the cursor forward by word rather than by character. The ESCAPE F command moves the cursor forward one word. Type:

{ESC}F

Notice that the cursor positions itself on the first blank space after the current word:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

When the ESCAPE F command is issued, the cursor moves forward to the first blank space it encounters or to the first punctuation mark it encounters after the current word.

Try it again. Type:

{ESC}F

three times and notice the cursor positions itself over the comma (,) following the word "hatched".

Your screen should look like this:

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:  
"The milk in this pail will provide me with cream,  
which I will make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now combine the ESCAPE <digit> command with an ESCAPE F command to move the cursor forward five words. Type:

```
{ESC}5{ESC}F
```

Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Notice the similarities between the ESCAPE and CONTROL letter commands. CONTROL-B moves the cursor back one character, while ESCAPE B moves it back one word. CONTROL-F moves the cursor forward one character and ESCAPE F moves the cursor forward one word.

The ESCAPE <digit> command repeats any of these commands as many times as you choose.

Moving the Cursor Up One Line

Although you can type ESCAPE B and ESCAPE F over and over again to move the cursor up a line or down a line, there is a more efficient way to move the cursor up and down. The CONTROL-Z command moves the cursor up one line. Try it. Type:

{CNTRL-Z}

Notice the new cursor position:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE



Now, type:

{ESC}3{CNTRL-Z}

The cursor moves up three lines and positions itself on the character directly above the previous cursor position:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now type:

{CNTRL-Z}

followed by:

{ESC}2{ESC}F

to position the cursor over the colon (:) at the end of the previous line:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Moving the Cursor Down One Line

The CONTROL-N command moves the cursor to the next line. Type:

{CNTRL-N}

The cursor moves down one line and positions itself on the character directly below the previous cursor position:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
 "The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now type:

{ESC}4{CNTRL-N}

followed by:

{ESC}6{ESC}F

to move the cursor to the period (.) following the word "poultry-yard":

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
 "The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard."

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now that you know how to move the cursor around on the screen, you can combine cursor commands with editing commands and begin to make changes to the practice paragraph.

#### COMMAND REVIEW

Here's a review of the commands you've learned so far:

CONTROL-B	Moves cursor backward one character
CONTROL-F	Moves cursor forward one character
CONTROL-N	Moves cursor down one line
CONTROL-Z	Moves cursor up one line
ESCAPE B	Moves cursor backward one word
ESCAPE F	Moves cursor forward one word
ESCAPE <digit>	Repeats command the number of times specified by <digit>

## MAKING CORRECTIONS TO TEXT

This section discusses how to make minor changes to text. Specifically, you will learn how to:

- Delete characters and words
- Insert characters and words

The more sophisticated editing commands, such as deleting entire lines and inserting lines and paragraphs, are discussed in Chapters 4 and 5.

### Deleting Characters

The CONTROL-D command deletes characters positioned under the cursor. Try it. Type:

```
{ESC}5{CNTRL-B}
```

to position the cursor over the hyphen (-). Then type:

```
{CNTRL-D}
```

and watch the hyphen disappear and the line rewrite itself on the screen. The screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultryard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Now, let's try two different methods to delete a word. First, try deleting a word character by character. Type:

{CNTRL-Z}

followed by:

{ESC}38{CNTRL-B}

to move the cursor up one line and over to the comma following the word "these". Your screen should look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Now type:

```
{CNTRL-D}
```

six times and watch the line shift to the left six times as it deletes the word "when" character by character. Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these hatched, will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Then type:

```
{ESC}9{CNTRL-D}
```

and watch the entire word "hatched," disappear and the line shift left only once.

Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Typing {CNTRL-D} six times and typing {ESC}9 {CNTRL-D} both delete one word. However, the ESCAPE 9 CONTROL-D command line is more efficient since it takes less time for EMACS to work.



You can use the {BACKSPACE} command to delete characters to the left of the cursor. Try it. Type:

```
{ESC}5{CNTRL-Z}{ESC}2{ESC}F
```

to move the cursor to the comma (,) after the word "along":

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Then type:

```
{BACKSPACE}
```

and watch the "g" disappear and the line rewrite itself.

#### Note

If your terminal has a DEL key, you can use it in place of {BACKSPACE}. The best key for you to use is the one that requires the least amount of typing.

Your screen should now look like this:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these will produce chickens, and by and by I shall have quite a large poultryyard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now combine an ESCAPE <digit> command with the {BACKSPACE} command to delete the rest of the word. Type:

```
{ESC}5{BACKSPACE}
```

The remainder of the word disappears, the line rewrites itself and your screen looks like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

### Deleting Words

Often, when editing, it will be more efficient for you to delete text word by word rather than character by character. You can use the ESCAPE D command to delete an entire word or part of a word. The ESCAPE D command works by deleting the character under the cursor and every character to the the right of the cursor until it reaches a blank space or a punctuation mark. Try a few examples using the ESCAPE D command so you can see how it works. Type:

```
{ESC}3{ESC}F
```

followed by:

```
{ESC}4{CNTRL-F}
```

to move the cursor up and over to the second "d" in "daydream".

Your screen should look like this:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these will produce chickens, and by and by I shall have quite a large poultryyard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now type:

{ESC}D

and watch what happens. The letters "dream" disappear. Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

If you want to use the ESCAPE D command to delete an entire word you must position the cursor on the blank space preceding the word. Try deleting the word "farmer's" from the first line. Type:

{ESC}3{CNTRL-Z}

followed by:

{ESC}6{ESC}B

followed by:

{CNTRL-B}

to move the cursor to the blank space preceding the word "farmer's":

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Now type:

```
{ESC}D
```

What happens? The ESCAPE D command deletes characters and stops when it reaches a punctuation mark or a blank space.

Your screen should look like this:

```
A's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

In this example, the cursor reaches a punctuation mark before it reaches the blank space at the end of the word. So, you still need to delete the rest of the word. Type another:

```
{ESC}D
```

and the rest of the word vanishes and the cursor stops on the blank space:

```
A daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

The ESCAPE {BACKSPACE} command is the opposite of the ESCAPE D command. ESCAPE {BACKSPACE} deletes all characters to the left of the cursor until it reaches a blank space or a punctuation mark. Type:

```
{ESC}6{ESC}F
```

to move the cursor to the end of the line.



Your screen should look like this:

```
A daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Now type:

```
{ESC}{BACKSPACE}
```

and notice what happens. The word "milk" is gone. The cursor is still positioned at the end of the line:

```
A daughter had been out to ■
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Try one more ESCAPE {BACKSPACE} command, but this time, combine it with an ESCAPE <digit> command. Type:

```
{CNTRL-N}
```

followed by:

```
{ESC}3{ESC}F
```

to move the cursor to the end of the next line. Now type:

```
{ESC}3{ESC}{BACKSPACE}
```

and notice the words "to the dairy" disappear. Your screen should now look like this:

```
A daughter had been out to
the cows, and was returning █
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

### Inserting Characters and Words

Let's review what you've done so far: you entered a paragraph, you moved the cursor around within the paragraph, and you deleted a few characters and words from the paragraph. Now it's time to insert the characters and words you've just deleted back into the paragraph. Step by step instructions follow.

When you're finished inserting the underlined text, the paragraph will read as follows:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultryyard.
```

You don't need to give EMACS any special commands when you want to insert text into an existing paragraph. All you do is position the cursor where you want the text to appear and type. The cursor marks the spot where text is inserted and moves to the right as each character you type appears on the screen.

When you insert text into an existing document, you can do so in whatever order you wish. You can start at the top of your document and work down; you can start at the end and work up; you can start in the middle and work up or down; or you can be totally unsystematic.

Your screen should currently look like this:

```
A daughter had been out to
the cows, and was returning █
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

The cursor is near the top of the paragraph, so let's begin at the top and work to the bottom.

Type:

```
{CNTRL-Z}{ESC}5{ESC}B
```

to move the cursor to the "d" in "daughter". You want to insert the word "farmer's" before the word "daughter". So just type:

```
farmer's
```

followed by a blank space. Your screen should now look like this:

```
A farmer's daughter had been out to
the cows, and was returning
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Now you want to add the word "milk" to the end of the current line. The cursor is still positioned over the "d" in "daughter". Were you to begin typing now, everything you type would be inserted before the "d". So, before you type the word "milk", you must move the cursor to the end of the line. Type:

```
{ESC}5{ESC}F
```

to move the cursor to the desired position. Now type one blank space followed by:

```
milk
```

Stop after you type the word "milk". Do not press the RETURN key or you'll insert a blank line.

Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Next, move the cursor down one line by typing:

```
{CNTRL-N}
```

Then type:

to the dairy

Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Now move the cursor down two lines and over the comma (,) by typing:

```
{ESC}2{CNTRL-N}
```

followed by:

```
{ESC}4{ESC}B
```

```
{ESC}2{CNTRL-B}
```

Then type a blank space followed by:

along

Your screen should now look like this:

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to day in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these will produce chickens, and by and by I shall have quite a large poultryyard.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

At the end of the current line you need to add the word "dream". Move the cursor to the end of the line by typing:

{ESC}4{ESC}F



Then type:

dream

so that your screen looks like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Five lines down, you need to insert the characters and words ", when hatched, ". Try this on your own. Move the cursor down five lines, position the cursor on the blank space before the word "will," and type:

, when hatched,

Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultryyard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Move the cursor down one line and forward to the "y" in "yard." When the cursor is in the correct position, type a hyphen:

Your screen should look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

You've now finished re-inserting the characters and words you deleted from your paragraph. If you left any typing errors when you entered the original paragraph, you can correct them now. All you need to do is position the cursor over the erroneous character, delete the erroneous character, then insert the correct character.

#### SAVING A DOCUMENT

You've just finished entering and editing a document. You're ready to exit EMACS. First, however, you must tell EMACS that you want to save your document. You must do this each time you exit EMACS, or all the changes you've made during your editing session will be lost.

The CONTROL-X CONTROL-S command saves the current document in the most recently named file. The file named "practice" is the only file you named during this EMACS session. So, when you use the CONTROL-X CONTROL-S command, your document is saved in the file named "practice".  
Type:

```
{CNTRL-X}{CNTRL-S}
```

In a few seconds, EMACS responds with a message indicating that it has saved your document in the file named "practice". It looks like this:

```
File written: practice
```

Now you're ready to exit EMACS.

### EXITING EMACS

There can be several different reasons for exiting EMACS. One reason is simply that you've finished your editing task or reached a suitable stopping point, you've saved your document in a file using the CONTROL-X CONTROL-S command, and now you're ready to return to PRIMOS command level. In this case, you use the CONTROL-X CONTROL-C command to exit EMACS. When you type:

```
{CNTRL-X}{CNTRL-C}
```

the system responds with the message:

```
OK,
```

You are now back at the PRIMOS command level. At this point, unless you have other work to do, you logout. If you want to restart EMACS and continue editing the file you just saved, see RESTARTING EMACS in Chapter 4 for instructions.

### Exiting EMACS Because of Mistakes

Another reason to exit EMACS is that something has gone wrong and you don't know how to recover. Don't panic. You're a beginner. You're bound to press the wrong key sooner or later, and you haven't learned enough about EMACS to know how to reverse your mistakes. In this case, you use the CONTROL-P command to exit EMACS.

At this stage in your EMACS career, you will probably resort to using the CONTROL-P command to recover from all types of disasters. It's okay to do so. However, if you've done a great deal of work and you use the CONTROL-P command to exit EMACS before you have saved your document in a file, your document is not saved. You will have to start over.

To use the CONTROL-P command, type:

```
{CNTRL-P}
```

The system responds with the message:

```
Type REN to abort command, START to continue.  
OK,
```

If you really do want to abandon the document you've been working on and start over, just ignore the message. You're at PRIMOS command level. You can return to STARTING EMACS at the beginning of this chapter and make a fresh start.

However, if you typed the CONTROL-P command by mistake, and you really don't want to stop EMACS, respond to:

Type REN to abort command, START to continue.  
OK,

by typing:

start{RETURN}

When you type START, EMACS returns your text to the screen and you can continue from where you left off. If your text does not reappear after a few seconds, type {CNTRL-L}. (See Chapter 6 for more details on the CONTROL-P command.)

## SUMMARY OF BASIC EMACS COMMANDS

In this chapter, you learned to use the following EMACS commands:

<u>Command</u>	<u>Description</u>
CONTROL-B	Moves cursor backward one character
ESCAPE B	Moves cursor backward one word
CONTROL-D	Deletes one character
ESCAPE D	Deletes one word
CONTROL-F	Moves cursor forward one character
ESCAPE F	Moves cursor forward one word
{BACKSPACE}	Deletes one character to the left of the cursor
ESCAPE {BACKSPACE}	Deletes one word to the left of the cursor
CONTROL-N	Moves cursor down one line
CONTROL-Z	Moves cursor up one line
CONTROL-X CONTROL-S	Saves text in a file
CONTROL-X CONTROL-F	Creates and names a file
CONTROL-X CONTROL-C	Returns user to PRIMOS command level
ESCAPE <digit>	Repeats EMACS command specified <NUMBER> of times

## CHAPTER 4

## MORE EMACS

This chapter explains an intermediate set of EMACS commands. It is structured like the previous chapter -- at least one example follows the explanation of each command. Try the examples as you read about each command. When you finish the examples in this chapter, you will know how to:

- Restart EMACS to edit an existing file
- Add new text to an existing file
- Move the cursor to the beginning and end of a line, to the beginning and end of a file, and backward and forward one screen at a time
- Search forward and backward through a file for a specific word or phrase

Remember, if you run into problems, you can use the CONTROL-P command described under EXITING EMACS in Chapter 3 or you can refer to Chapter 6 for help.

## RESTARTING EMACS

In the previous chapter, you entered a paragraph of text, edited the text a bit, and then saved the text in a file named "practice". Now, restart EMACS and tell it you want to edit the file named "practice". There are two ways you can do this. Before you try either method, though, be sure you are logged into the directory containing your file "practice". (If you are not sure how to log into a directory, see Chapter 2 of the New User's Guide to EDITOR and RUNOFF.)

Once you're in the correct directory, you can type:

```
EMACS -TTP <terminal type> PRACTICE {RETURN}
```

which tells EMACS you want to edit the existing file named "practice". After you've typed this command line, EMACS will automatically display the file "practice" on your screen.

Or, you can type:

```
EMACS -TTP <terminal type> {RETURN}
```

and then issue the CONTROL-X CONTROL-F command once EMACS is running, just as you did in Chapter 3.

When you type:

```
{CNTRL-X}{CNTRL-F}
```

EMACS responds by printing the words:

Find file:

When you see this prompt, go ahead and type:

```
practice{RETURN}
```

No matter which method you use, your screen will look like this in a few seconds:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to day
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```



Now, type:

```
{ESC}X LOAD {RETURN}
```

In a few seconds, you will see the message:

```
Autoload complete
```

EMACS is now ready and waiting for you to tell it what to do.

#### Note

You've just gone through the steps necessary to start EMACS, find an existing file, and load the special libraries. From now on, you'll be expected to follow this procedure each time you start EMACS.

#### ADDING TEXT AND MOVING THE CURSOR

When working on a document, you will often want to add new text to it. To add text to the end of a document, all you do is position the cursor at the end of the file and type. To add text to the beginning or middle of a document, you need to position the cursor at the appropriate character, and then type the new text.

Let's begin by adding a new paragraph to the end of the document.

Moving the Cursor to the End of a File

Take a look at your screen and notice the current cursor position. Since you want to add text to the end of your document, you need to move the cursor from its current position to the end of the file. The ESCAPE > command is the easiest way to move the cursor from anywhere in the file to the end of the file. Try it. Type:

{ESC}>

(Remember to use the SHIFT key when you type >.) Notice the new cursor position:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.

■

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Adding Text to the End of a File

To add text to the end of a file, all you need to do is type, so go ahead and type the following paragraph. As you type, watch what happens when you run out of lines on your screen. When you fill up a screen, EMACS automatically starts a new screen for you. This new screen contains a few lines from the previous screen plus blank lines. You do not have to stop typing while EMACS creates a new screen. EMACS remembers what you type and adds it to the new screen.

Type:

```

*****{RETURN}
And all the young fellows will admire it,{RETURN}
and come and make love to me,{RETURN}
but I shall toss my head and have nothing{RETURN}
to say to them."{RETURN}
Forgetting all about the pail,{RETURN}
and suiting the action to the word,{RETURN}
she tossed her head.{RETURN}
Down went the pail, all the milk{RETURN}
was spilled, and all her fine castles{RETURN}
in the air vanished in a moment!{RETURN}
Moral: Do not count your chickens{RETURN}
before they are hatched.

```

Your screen should now look like this:

```

And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.

```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

### Moving the Cursor to the Beginning of a File

Most of your documents will be too long to fit on one screen. So, EMACS has a few commands that make it easy for you to move around within a large file. You already know that the ESCAPE > command moves the cursor to the end of a file. If you want to move the cursor from anywhere in a file to the beginning of a file, use the ESCAPE < command. Try it now. Type:

```
{ESC}<
```

(Remember to depress the SHIFT key when you type <.) The cursor is now positioned at the beginning of the file:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

### Moving the Cursor Forward One Screen

EMACS has a command that enables you to move the cursor forward screen by screen and a command that enables you to move the cursor backward screen by screen. The CONTROL-V command moves the cursor forward one screen. This command is very handy for reading through a file. Try it now. Type:

```
{CNTRL-V}
```

Your screen should now look like this:

```
And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."  
Forgetting all about the pail,  
and suiting the action to the word,  
she tossed her head.  
Down went the pail, all the milk  
was spilled, and all her fine castles  
in the air vanished in a moment!  
Moral: Do not count your chickens  
before they are hatched.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Notice where the cursor positions itself on the new screen. Also, notice that EMACS overlaps a few lines from the previous screen onto the new screen. If the current document were longer, you could continue to use CONTROL-V commands to view the entire file.

Note that as you move to each new screen, EMACS normally places the cursor in the middle of the screen. When the text in a file does not extend to the middle of the screen, EMACS places the cursor at the end of the document.

#### Moving the Cursor Backward One Screen

The ESCAPE V command moves the cursor backward one screen. To try it, type:

```
{ESC}V
```

Your screen should now look like this:

```

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Notice where the cursor positions itself in the new screen.

### Moving the Cursor Forward to a Specific Word

Suppose you're working on a long document and you need to find a specific character, word, or phrase. Certainly, you wouldn't want to scroll through the file screen by screen. It would take too long. Instead, you can use the CONTROL-S command.

#### Note

On some terminals you must use the ESCAPE S command in place of CONTROL-S. If CONTROL-S does not work when you try it, substitute ESCAPE S for CONTROL-S in the following examples.

The CONTROL-S command searches forward through a file and moves the cursor to the first occurrence of the character, word, or phrase you've specified. Try it. Use the ESC < command to move the cursor to the beginning of the file. Then type:

```
{CNTRL-S}
```

EMACS responds to the CONTROL-S command by printing the words:

Forward Search:

below the mode line:

```

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
Forward Search:

```

EMACS wants you to type in the character, word, or phrase that you want to search for. You must type the character, word, or phrase exactly as it appears in your document or EMACS won't find it. For instance, if you forgot to type in a capital letter and the word you're searching for has one, EMACS won't find it. Also, if you are searching for a phrase, all of the words in the phrase must be on one line in the document or EMACS won't find it.

Let's search for the word "chickens" but let's make a typing error the first time we try so we can see what happens.

Type:

```
chikens{RETURN}
```

EMACS returns the message:

```
Not found: chikens
```

under the mode line. Now you need to issue the CONTROL-S command again. So, type:

```
{CNTRL-S}
```

This time respond to the "Forward Search:" request by typing:

```
chickens{RETURN}
```

The cursor moves forward to the first occurrence of the word "chickens" following the current cursor position and stops over the comma (,) at the end of the word:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Try the exact same command one more time and save yourself some typing while you do it. The CONTROL-C command re-executes the last EMACS command you typed. Try it. Type:

```
{CNTRL-C}
```

The cursor moves forward to the next occurrence of the word "chickens" following the current cursor position and stops over the blank space at the end of the word.



Your screen should look like this:

```

And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.

```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

### Moving the Cursor Backward to a Specific Word

The CONTROL-R command searches backward through a file for the character, word or phrase you specify. Try searching backward for the character "\*" which is somewhere in the middle of the file. Type:

```
{CNTRL-R}
```

EMACS responds by printing the words:

Reverse Search:

below the mode line.

You want to find the character "\*". So, type:

```
*{RETURN}
```

Notice that the cursor moves backward and stops over the first "\*" that it finds:

```

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

### Adding Text to the Middle and Beginning of a File

When you need to add text to the middle or to the beginning of a document, you have two choices: you can simply type {RETURN} after each line or you can use the CONTROL-O command to insert blank lines that you can fill up later as you type. Let's try both ways. Whichever method you choose to use in the future is up to you.

Let's add several new lines of text to the beginning of the second paragraph without first making room for the new lines. Step by step instructions follow. When you're finished, the beginning of the paragraph should read as follows:

```

Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.

```

To begin, use the CONTROL-F command to position the cursor on the blank space at the end of the current line. Now type:

```
{RETURN}
```

Notice what happens when you depress the RETURN key. The cursor moves down to the beginning of a blank line and the text below it moves down to make room for this blank line. Your screen should now look like this:

```
A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
```

```
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

When you type in the remainder of the new text, EMACS inserts a blank line and rewrites the screen every time you depress the RETURN key. Type:

```
Then I shall sell some of my fowls{RETURN}
and with the money which they will bring in{RETURN}
I will buy myself a new gown,{RETURN}
which I shall wear when I go to the fair.
```

Your screen should now look like this:

```

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Some people find the constant rewriting of the screen as they add new text very annoying. If you're one of these people, you can use the ESCAPE <digit> CONTROL-O command to insert blank lines. The screen rewrites itself only once to accommodate the blank lines. Then you can type the new text on the blank lines. Try it.

Let's add some text to the beginning of the document. Type:

```
{ESC}<
```

to move the cursor to the beginning of the file:

```

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Now type:

```
{ESC}10{CNTRL-O}
```

to insert 10 blank lines. Notice the screen rewrites itself only once when inserting the ten blank lines.

Your screen should now look like this:

```

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Let's add a title line and an author line to the document. Type 10 spaces followed by:

```
THE MILKMAID AND HER PAIL{CNTRL-N}
```

Now type 15 spaces followed by:

```
from Aesop's Fables{CNTRL-N}
```

Your screen should now look like this:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:  
"The milk in this pail will provide me with cream,  
which I will make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

We have no more text to add to the remaining blank lines, so let's  
delete all but two of them. Type:

{CNTRL-D}

enough times to leave only two blank lines after the current one.

Your screen should now look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Notice that CONTROL-D was used here to delete blank lines, even though its normal function is to delete only one character. This command worked here because a line with no words or spaces on it contains a special line separator visible only to EMACS. This separator is one character wide. When CONTROL-D deletes the separator, EMACS no longer holds the space for a line.

#### Moving the Cursor to the Beginning of a Line

The CONTROL-A command moves the cursor from its current position to the beginning of the line. This command is a true time saver. Try it. Position the cursor in the middle of a line anywhere on your screen and type:

```
{CNTRL-A}
```

The cursor moves in one jump from its current position on the line to the first character on the line.

#### Moving the Cursor to the End of a Line

The CONTROL-E command moves the cursor from its current position to the end of the line. Try it. Type:

```
{CNTRL-E}
```



The cursor moves in one jump from its current position on the line to the blank space at the end of the line.

#### COMMAND SUMMARY

In this chapter, you learned to use the following EMACS commands:

<u>Command</u>	<u>Description</u>
CONTROL-A	Moves cursor to beginning of current line
CONTROL-C	Re-executes previous command
CONTROL-E	Moves cursor to end of current line
CONTROL-O	Inserts blank line
CONTROL-R	Searches backward for a specific word
CONTROL-S	Searches forward for a specific word
CONTROL-V	Moves cursor forward one screen
ESCAPE V	Moves cursor backward one screen
ESCAPE <	Moves cursor to beginning of file
ESCAPE >	Moves cursor to end of file
CONTROL-X CONTROL-F	Finds an existing file

## CHAPTER 5

## MOVING TEXT

This chapter explains several commands that are useful for moving text around within a document. When you have mastered these commands you will have only scratched the surface of the text-moving capabilities.

When you finish the examples in this chapter, you will know how to:

- Delete lines of text
- Delete specified regions of text
- Yank text
- Set and switch the mark and point
- Move text
- Erase incorrectly yanked text

## DELETING AND YANKING LINES OF TEXT

In Chapter 3, you deleted characters and words with the CONTROL-D and the ESCAPE D commands; in Chapter 4, you deleted blank lines with the CONTROL-D command. To delete or "kill" entire lines of text, you use the CONTROL-K command.

In EMACS, a line of text contains characters plus the invisible line separator at the end of the line. The CONTROL-K command deletes text on a line from the cursor position to the line separator. This means: if you want to delete part of a line of text, you must position the cursor over the first character you want to delete and then type the CONTROL-K command; if you want to delete an entire line of text, you must position the cursor at the beginning of the line and then type the CONTROL-K command twice. The first CONTROL-K kills the characters on the line. The second CONTROL-K kills the line separator. Let's try a few CONTROL-K commands.

Use any method you want to move the cursor to the "p" in the word "provide." Your screen should look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Now type:

```
{CNTRL-K}
```

Notice that the text from the cursor position to the line separator disappears. The invisible line separator remains at the end of the line. The cursor remains in its current position.

Your screen should now look like this:

```

                THE MILKMAID AND HER PAIL
                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Whenever you delete more than one character of text, EMACS takes the deleted text and places it in storage. If you need to, you can use the CONTROL-Y command to yank (reinsert) the most recently deleted text into your document at the cursor. Yank the text you just deleted. Type:

```
{CNTRL-Y}
```

The deleted text reappears in front of the cursor. The cursor remains over the blank space at the end of the line.

Your screen should look like this:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail will provide me with cream,  
which I will make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

Then I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now use the CONTROL-N command followed by the CONTROL-A command to move the cursor to the beginning of the next line:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail will provide me with cream,  
which I will make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

Then I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Then type:

{CNTRL-K}

The entire line of characters disappears. A blank line containing the invisible line separator remains. Your screen should look like this:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:  
"The milk in this pail will provide me with cream,  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

Then I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Type:

{CNTRL-K}

to delete the line separator. The screen rewrites itself to fill in the blank line:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail will provide me with cream,  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

Then I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

The ESCAPE 2 CONTROL-K command deletes a line of characters plus the line separator. Try it. Type:

{ESC}2{CNTRL-K}

The entire line disappears and the screen rewrites itself.



Your screen should now look like this:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail will provide me with cream,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

Then I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

When you use an ESCAPE <digit> command with the CONTROL-K command to delete more than one line, EMACS automatically deletes all the line separators after the first one. Therefore, it takes two CONTROL-K commands to delete the first line, but only one CONTROL-K command to delete each remaining line and its line separator.

This means that if you want to delete three lines of text, you need two CONTROL-K commands to delete the first line, one CONTROL-K command to delete the second line, and one CONTROL-K command to delete the third line. You need a total of four CONTROL-K commands to delete three lines of text.

Try it. Type:

{ESC}4{CNTRL-K}

Three lines of text disappear and the screen rewrites itself to look like this:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:  
"The milk in this pail will provide me with cream,  
~~Then~~ I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."  
Forgetting all about the pail,  
and suiting the action to the word,  
she tossed her head.  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

EMACS saves text that is deleted with consecutive delete or kill commands together in a block. This means that you can use the CONTROL-Y command to yank the five lines you just deleted with consecutive CONTROL-K commands. Try it. Type:

{CNTRL-Y}

The five lines reappear. Your screen should now look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

The ESCAPE 0 CONTROL-K command deletes text from the cursor to the beginning of the line. Try it.

Move the cursor over two characters using the CONTROL-F command. Then type:

```
{ESC}0{CNTRL-K}
```

The two characters to the left of the cursor disappear and the line rewrites itself. Your screen should now look like this:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail will provide me with cream,  
which I will make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

Then I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

You just deleted two characters. Now yank them back by typing:

{CNTRL-Y}

Your screen should now look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

So far in this chapter, you have learned that:

- Whenever you delete more than one character, EMACS saves the deleted text.
- You can use the CONTROL-Y command to reinsert the most recently deleted text at the cursor.
- When you use any number of delete commands consecutively, EMACS saves the text together in one block.
- You can use the CONTROL-Y command to reinsert at the cursor the most recently deleted block of text.

#### DELETING AND YANKING REGIONS OF TEXT

So far, you have learned to delete characters, words, partial lines, and entire lines of text using variations of the CONTROL-D, ESCAPE D, and CONTROL-K commands. You've also learned to use the CONTROL-Y command to yank the most recently deleted text and to reinsert that text into your document at the cursor. You're now ready to learn to delete and yank large portions of text called regions.

Deleting regions of text takes two steps:

1. First, you put a boundary around the text you want to delete.
2. Second, you issue the CONTROL-W command.

You put a boundary around text by setting the mark and the point. To set the mark, move the cursor to the beginning of the region you want to delete and then issue the CONTROL-@ command. Let's try it. Use the CONTROL-R command and search for the word "make". (If you don't remember how to use it, see Chapter 4.) Place the cursor on the "m":

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*  
Then I shall sell some of my fowls, and with the money which they will bring in I will buy myself a new gown, which I shall wear when I go to the fair. And all the young fellows will admire it, and come and make love to me,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now set the mark at the cursor by typing:

{CNTRL-@}

Notice that the mark is invisible. Your screen looks exactly the same as it did before you set the mark.

Now use the CONTROL-S command and search for the word "sell" followed by the CONTROL-B command to backup four spaces and position the cursor on the "s":

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*  
Then I shall sell some of my fowls, and with the money which they will bring in I will buy myself a new gown, which I shall wear when I go to the fair. And all the young fellows will admire it, and come and make love to me,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

You now have a region of text that is bounded by the invisible mark and the cursor. The cursor is the point. You don't have to issue a command to set it. Just leave the cursor where it is. Since the mark and point are set, you've defined a region. Now type:

{CTRL-W}

to delete the region.

Your screen should look like this:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail will provide me with cream,  
which I will sell. With the money I will buy a number of eggs,  
and these, when hatched, will produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

Then I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

The CONTROL-Y command yanks the most recently deleted region. Type:

{CNTRL-Y}

and the deleted region reinserts into your document at the cursor.



Now delete one more region. Move the cursor forward to the "F" in the word "Forgetting." Type:

```
{CNTRL-@}
```

to set the mark. Now use the CONTROL-S command and search for the character "!":

```
*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

### Switching the Mark and Point

Since the cursor marks the point, you now have a region bounded by an invisible mark and the point. Where's the invisible mark? If you can't remember type:

```
{CNTRL-X}{CNTRL-X}
```

and the point and mark switch places. If you issue the CONTROL-W command, you will delete the region from the "F" (which is now the point) to the "!" (which is now the mark). Do it. Type:

```
{CNTRL-W}
```

The region should disappear and your screen should look like this:

```
*****  
And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."
```

```
█  
Moral: Do not count your chickens  
before they are hatched.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

EMACS saves the region you just deleted. Until you delete more text, you can yank back this region with the CONTROL-Y command. Type:

```
{CNTRL-Y}
```

The deleted region reinserts into your document at the cursor:

```
*****  
And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."  
Forgetting all about the pail,  
and suiting the action to the word,  
she tossed her head.  
Down went the pail, all the milk  
was spilled, and all her fine castles  
in the air vanished in a moment!  
Moral: Do not count your chickens  
before they are hatched.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

When you yank the most recently deleted text from storage, you are actually yanking a copy of the text. The text remains in storage; consequently, you can use the CONTROL-Y command to yank the most recently deleted text as many times as you want.

#### MOVING TEXT

So far, you've always yanked deleted text back to its original location in the document. Let's move the cursor from its current position to another location and yank the most recently deleted region at the new cursor position.

Use the CONTROL-R command and search backward for the characters "\*\*\*\*\*":

```
*****  
And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."  
Forgetting all about the pail,  
and suiting the action to the word,  
she tossed her head.  
Down went the pail, all the milk  
was spilled, and all her fine castles  
in the air vanished in a moment!  
Moral: Do not count your chickens  
before they are hatched.
```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Now type:

```
{CNTRL-Y}
```

to yank the most recently deleted region and insert it into the document at the cursor.

Your screen should now look something like this:

```

Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.

```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

#### Note

Do not worry if the text on your screen is not displayed exactly like the examples you see in this book. From now on, EMACS may display your text differently from time to time depending on the command sequence you use. The position of the cursor, however, should be the same.

When you're working on a document, you can repeat the process of repositioning the cursor and yanking the same text as many times as you need to. Remember, however, that a word, line, or region of text is only available for yanking with the CONTROL-Y command until you delete another word, line, or region. Therefore, when you move text around within your document, it is wisest to reposition the cursor and issue the CONTROL-Y command immediately after deleting the text you want to move.

#### ERASING INCORRECTLY YANKED TEXT

If you yank deleted text and find that you either reinserted it into the document in the wrong place or that you yanked the wrong text, you can use the CONTROL-W command to erase the yanked text. Try it now. Type:

{CTRL-W}

to erase the region of deleted text you just yanked.

Your screen should now look this:

```

*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.

```

```
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

## COMMAND SUMMARY

In this chapter, you learned to use the following EMACS commands:

<u>Command</u>	<u>Description</u>
CONTROL-K	Deletes lines and partial lines of text
CONTROL-W	Deletes regions of text; erases incorrectly yanked text
CONTROL-Y	Yanks most recently deleted text from storage and reinserts it into document at cursor
CONTROL-@	Sets mark at cursor
CONTROL-X CONTROL-X	Switches mark and point

## CHAPTER 6

## HELP AND HELPFUL HINTS

This chapter will help you handle specific situations that may occur when you use EMACS.

This chapter does not contain examples for you to try. There are a few screens displayed in this chapter. These screens are included to demonstrate how certain commands affect your screen when you use them.

Issues discussed in this chapter include:

- Aborting EMACS commands
- Refreshing your terminal screen
- Help commands
- Displaying previous EMACS commands
- Remembering EMACS commands
- Responding to specific situations

## ABORTING RUNNING EMACS COMMANDS

The CONTROL-P command aborts running EMACS commands. For example, if you issue the CONTROL-V command, you can abort that command while EMACS writes the next screen by typing:

```
{CNTRL-P}
```

The CONTROL-P command returns you to PRIMOS command level and issues the prompt:

Type REN to abort command, START to continue.

If you ignore this prompt and type something else, you will remain at PRIMOS command level and any changes you made in EMACS will be lost. If you type:

```
ren{RETURN}
```

or

```
start{RETURN}
```

you are returned to EMACS. REN ignores the command you typed just before {CNTRL-P} and leaves your screen unchanged. START reexecutes

the command you typed just before {CNTRL-P} and changes your screen to reflect that command.

When you type either of these commands, your screen may be blank or jumbled when you return to EMACS. If this happens, you can use the CONTROL-L command to refresh your screen. It should make your text reappear in its normal state.

#### ABORTING PARTIALLY INVOKED EMACS COMMANDS

The CONTROL-G command aborts partially invoked EMACS commands. If you issue (either by mistake or on purpose) any of the EMACS commands that prompt you for a response, you can choose not to respond and abort that command by typing:

{CNTRL-G}

For example, you can type CONTROL-G to abort the CONTROL-S command rather than respond to its "Forward Search:" prompt.

If you inadvertently hold down the ESCAPE key too long, EMACS thinks you are trying to issue more than one ESCAPE command. When this happens, EMACS prompts with:

PL:

or

<number>PL:

on the line below the status line. If this happens to you, type:

{CNTRL-G}

as many times as necessary until the prompt disappears.

#### REFRESHING THE SCREEN

Occasionally, while in EMACS, you receive messages from PRIMDS. These messages print across the middle of your screen overwriting text as they print. When this happens to you, you can use the CONTROL-L command to clear and redisplay (refresh) your screen.



## HELP COMMANDS

To find out what any command does or which command to use to perform some task, you can use the CONTROL-\_ command. It is the basic help command in EMACS and it gives you different types of information about EMACS commands.

When you type CONTROL-\_, EMACS responds with this prompt:

Help on tap: C, A, D, L, or ?

which lists the command options you now have. These are discussed below.

Finding Out What a Command Does

If you are thinking about using a certain EMACS command, but forget exactly what the command does, use the C option. When you type:

{CNTRL-}\_C

EMACS prompts with the words:

Explain key

below the mode line, as shown below:

■ THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:  
"The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

Then I shall sell some of my fowls,  
and with the money which they will bring in  
I will buy myself a new gown,  
which I shall wear when I go to the fair.  
And all the young fellows will admire it,  
and come and make love to me,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Explain key

EMACS wants you to type the EMACS command you need explained. For example, if you want the CONTROL-D command explained, simply type:

```
{CNTRL-D}
```

and EMACS responds with an explanation of the command under the mode line.

Your screen should look like this:

```

      ■      THE MILKMAID AND HER PAIL
              from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

delete_char: Delete character

```

This will work for any combination of keystrokes. If you type in something that EMACS doesn't recognize, EMACS will usually prompt you with the message:

```
undefined key
```

### Finding Out Which Commands to Use

If you forget which EMACS commands you can use to perform a certain task, the A option can help you. When you type:

```
{CNTRL-}_A
```

EMACS responds with the prompt:

Apropos:

under the mode line and waits for you to specify a subject. When you do, EMACS searches through its library of commands and prints a list of commands related to that subject at the top of the screen. For example, suppose you've forgotten which command to use to search forward for a word. Just type:

```
search{RETURN}
```

and EMACS prints information about the commands that search at the top of your screen, separated by a double row of dashes and the cursor:

```
*Main command definitions*
^R      reverse_search_command  Reverse Search
^S      ^s_forward_search_command Forward Search {May stop output}
esc-R   reverse_search_command  Reverse Search
esc-S   forward_search_command  Forward Search
esc-r   reverse_search_command  Reverse Search
esc-s   forward_search_command  Forward Search
```

in the following manner:

```
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
```

```
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair.
And all the young fellows will admire it,
and come and make love to me,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Even though the information about the commands prints over the text that is already on your screen, this information does not become part of your document. After reading the information, you can use the CONTROL-L command to redisplay your original screen.

If EMACS cannot find information about the subject you've specified, it will respond with only the dashes at the top of your screen. When this happens, don't give up. Try another word related to your subject, or even a portion of the first word you tried. For instance, if you had typed "searching" in the previous example, it would not have worked. Typing only "search", however, was acceptable to EMACS and the information you needed was provided.

### Finding Out What You've Typed

Sometimes you may make a typing error while issuing an EMACS command. Your typing error might result in a command that EMACS doesn't recognize. In this case, EMACS does nothing to your document and you can retype the command. Or, your typing error might result in a command that EMACS does recognize and carries out. If this happens and you aren't sure which command you accidentally typed that caused unexpected changes to occur in your document, you can use the L option to find out. It prints the last 20 characters you typed. When you type:

```
{CNTRL-_L}
```

EMACS prints a list of the last 20 keystrokes you typed under the mode line. Note: EMACS uses the caret symbol (^) to signify the CONTROL key and the notation "esc" to signify the ESCAPE key when it lists the keystrokes.

When you type this command, your screen will look similar to this:

```

      THE MILKMAID AND HER PAIL
      from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
Then I shall sell some of my fowls,
and with the money which they will bring in
I will buy myself a new gown,
which I shall wear when I go to the fair
And all the young fellows will admire it,
and come and make love to me,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

-esc-v^z^z-bs^D^V-l^Gtwo-esc-v.^K^z^E-lf^X^IA-lf

```

If EMACS prints a command you don't recognize, you can assume it's the accidentally typed command. You can then use the CONTROL-C command to find out what this command does. Once you know how the accidentally typed command affected your document, you can fix any damage it might have done.

### Other Information

The D option of the CONTROL-\_ command describes EMACS commands related to a particular subject in detail. Since it is primarily useful for writing EMACS extensions, it will probably not be helpful to you until you are very familiar with EMACS. See the EMACS Reference Guide for information about this option.

The ? option of the CONTROL-\_ command simply prints a list of what all the other options do at the top of the screen in case you've forgotten. Therefore, the only help command you really have to remember is CONTROL-\_, since it lists all the other command options and what they do for you.

## REMEMBERING COMMANDS

If you've been thinking that there are a lot of EMACS commands to learn, you are right. However, remembering commands is not as difficult as it may seem at first.

With a few exceptions, EMACS commands are named to help you remember what they do. For example, CONTROL-D and ESCAPE D delete characters or text. CONTROL-N moves the cursor to the next line. CONTROL-Y yanks back previously delete text.

Once you've had some practice with EMACS, you'll remember the commands that you use most often. If you forget what a command does, how to do something, or commands you've just typed, you can use one of the help commands to refresh your memory. Additionally, Appendix B contains a complete listing of EMACS commands.

## RESPONDING TO SPECIFIC SITUATIONS

- Your screen goes blank: Try the CONTROL-L command. If that doesn't work, use the CONTROL-P command.
- A system message writes across your screen: Use the CONTROL-L command.
- Your screen gets garbled: Use the CONTROL-L command.
- The system crashes before you've saved your document: Start over. (See Chapter 3.)
- You exit EMACS before you've saved your document: Start over. (See Chapter 3.)
- You want to use a help command, but you can't remember which one does what you want: Use the CONTROL-\_ command.
- You mistype an EMACS command and strange things happen to your document: Use the CONTROL-\_L command.
- You forget what a specific command does: Use the CONTROL-\_C command.
- You type an EMACS command and the message "Invalid command" appears under the mode line: Type ESCAPE X LOAD. (See Chapter 3.)

## COMMAND SUMMARY

In this chapter, you learned about the following EMACS commands:

<u>Command</u>	<u>Description</u>
CONTROL-G	Aborts current EMACS command
CONTROL-P	Aborts running EMACS command
CONTROL-_	Basic help command
CONTROL-_A	Lists EMACS commands by function
CONTROL-_C	Explains EMACS command you specify
CONTROL-_L	Lists last 20 characters you typed

## CHAPTER 7

## REPLACEMENT COMMANDS

This chapter describes how to replace one string of characters in your file with another string of characters automatically.

Before you begin replacement, you must define a region. Let's do that now. Use any method you like to position the cursor on the word "A" at the beginning of the first paragraph of your text. Type:

```
{CNTRL-@}
```

to set the mark at the present cursor position. (See Chapter 5 if you've forgotten what this command does and how it works.) Your screen will not change. Now use any method you like to position the cursor on the period after the word "poultry-yard". You've defined a region. Your screen should look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I will make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, will produce chickens,
and by and by I shall have quite a large poultry-yard■
*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

To be sure that the mark is where you want it, type:

```
{CNTRL-X}{CNTRL-X}
```

to switch the mark and the point. The cursor should be at the beginning of the first paragraph.



Now type:

```
{CNTRL-X}{CNTRL-X}
```

again to reposition the cursor. You are ready to begin replacement.

#### SIMPLE REPLACEMENT

Suppose you want to replace each occurrence of "will" in the first paragraph with "would." You could look for each occurrence yourself, type in the new word, and then delete the old one, but that would take a lot of time. The ESCAPE X Replace command will do all this for you in one step. Let's try it. Type:

```
{ESC}X replace{RETURN}
```

EMACS will respond with the prompt:

```
Search for:
```

below the mode line. In response type:

```
will{Return}
```

EMACS will then print:

```
Replace with:
```

Now type:

```
would{RETURN}
```

Watch EMACS change your file automatically to:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:

"The milk in this pail would provide me with cream, which I would make into butter and take to market to sell. With the money I would buy a number of eggs, and these, when hatched, would produce chickens, and by and by I shall have quite a large poultry-yard. ■  
\*\*\*\*\*

And all the young fellows will admire it, and come and make love to me, but I shall toss my head and have nothing to say to them."

Forgetting all about the pail, and suiting the action to the word,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Each occurrence of "will" in the first paragraph is now replaced with "would". Notice that although the word "will" appears in the second paragraph, it was not replaced. The ESCAPE X Replace command works only within your defined region.

Let's try this command again. Type:

```
{ESC}X Replace{RETURN}
```

Followed by:

```
will{RETURN}
```

Followed by:

```
would{RETURN}
```

Since the word "will" is no longer present in the region, EMACS responds with the error message:

```
Not found: will
```

below the mode line.

## QUERY REPLACEMENT

Sometimes you will want to replace some instances of a frequently occurring string of characters but not all of them. The ESCAPE % command helps you do this. It is called the "Query Replace" command. Try it now. Type:

```
{ESC}%
```

EMACS will respond with:

```
Search for:
```

below the mode line, just as it did in the ESCAPE X Replace command. Now type:

```
would{RETURN}
```

EMACS will print:

```
Replace with:
```

Type:

```
will{RETURN}
```

EMACS will now move the cursor to the space after the first occurrence of "would" in the region and wait for a response from you. Your screen should look like this:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail would provide me with cream,  
which I would make into butter and take to market  
to sell. With the money I would buy a number of eggs,  
and these, when hatched, would produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."

Forgetting all about the pail,  
and suiting the action to the word,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Type ^G, CR, space, or "."

Notice that EMACS prints this message below the mode line:

Type ^G, CR, space or "."

These are the commands that you can use to tell EMACS what to do when you use this command.

Let's go ahead and replace this "would" with "will". To tell EMACS to do this, depress the space bar:

{Space}

EMACS will make the replacement and find the next "would" in the region.

Your screen should look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I would make into butter and take to market
to sell. With the money I would buy a number of eggs,
and these, when hatched, would produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

Type ^G, CR, space, or "."

```

EMACS is waiting for your response. This time, we won't make the replacement. To tell EMACS, type:

{RETURN}

EMACS will leave this "would" alone and go to the next one.

Again, EMACS waits for your response. Suppose you want to replace this "would" but you know it's the last change you want to make. You want to tell EMACS to make one last replacement and then stop searching. To do this, type a period:

EMACS makes the replacement and returns the cursor to where it was before you issued the ESCAPE % command.

Your screen should look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
"The milk in this pail will provide me with cream,
which I would make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, would produce chickens,
and by and by I shall have quite a large poultry-yard.
*****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
EMACS 714 (practice) * <MISCEL>LILLI>PRACTICE

```

If you want to stop the replacement process at any point once it is started, you can issue the `CONTROL-G` command. Once you type it, `EMACS` stops searching, makes no more replacements, and returns the cursor to its original position.

Keep in mind that as long as the `ESCAPE %` command is in effect, `EMACS` ignores any responses you give it except for the four commands discussed above.

## COMMAND SUMMARY

The replacement commands work like this:

<u>Command</u>	<u>Description</u>
ESCAPE X Replace (original word) {RETURN} (new word) {RETURN}	Replaces every instance of the original word with the new word within the specified region.
ESCAPE %	Finds each instance of the original word within the specified region and waits for you to decide whether or not to replace it.
{SPACE}	Replaces original word with new word.
{RETURN}	Skips to next occurrence of original word without replacing current one.
	Replaces original word currently at cursor and then stops replacement search.
CONTROL-G	Does no more replacement and stops search.



## CHAPTER 8

## FORMATTING COMMANDS

This chapter describes some EMACS commands that are useful for formatting documents. When you've finished this chapter, you'll be able to:

- Indent lines of text
- Center lines of text
- Use tabs
- Fill paragraphs evenly

## INDENTING LINES OF TEXT

To indent a single line of text in EMACS, you could position the cursor at the beginning of the line and type in the number of spaces you want to indent. If you have many lines to indent, however, you may find it tedious to do all that typing. The ESCAPE I command can save you some keystrokes. Let's try it.

Use any command you like to position the cursor on the first double quotation mark (") in your text:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she began to daydream in the following manner:

■The milk in this pail will provide me with cream, which I would make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, would produce chickens, and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

And all the young fellows will admire it, and come and make love to me, but I shall toss my head and have nothing to say to them."

Forgetting all about the pail,  
and suiting the action to the word,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Now type:

```
{ESC}5 {SPACE}
```

to indent the line five spaces:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    The milk in this pail will provide me with cream,
which I would make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, would produce chickens,
and by and by I shall have quite a large poultry-yard.
****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Now position the cursor anywhere on the following line. Type:

```
{ESC}I
```

#### Note

If you see the message:

```
Invalid command
```

under the mode line after you type {ESC}I, do not worry. You just forgot to type the LOAD command when you started EMACS. After you type:

```
{ESC}X LOAD{RETURN}
```

The ESCAPE I command will work as described.

The line will now be indented just like the one above it:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk  
the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail will provide me with cream,  
which I would make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, would produce chickens,  
and by and by I shall have quite a large poultry-yard.  
\*\*\*\*\*

And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."

Forgetting all about the pail,  
and suiting the action to the word,  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

#### CENTERING A LINE

The CONTROL-X CONTROL-Z S command centers a line of text on your screen relative to a 70 character line (the size of your screen.) To use it, just position the cursor anywhere on the line you want to center and then type the command. Try it now. Position the cursor somewhere on the line of stars (\*) in your text. Type:

{CNTRL-X}{CNTRL-Z}S

The stars should now be centered on the screen like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
      which I would make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, would produce chickens,
and by and by I shall have quite a large poultry-yard.
                                *****
and all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

If you have more than one consecutive line to center, you can use the ESCAPE <digit> command to repeat the execution of CONTROL-X CONTROL-Z S. To try it, position the cursor somewhere on the title line of the text. Type:

```
{ESC}2 {CNTRL-X}{CNTRL-Z}S
```

Both the title line and the one below it will be centered on your screen. The cursor will move to the following line.

#### USING TABS

There are several ways to set tabs in EMACS. The method you use will depend on your task.



EMACS will respond with:

How far apart:

Type:

8{RETURN}

The prompt message will disappear. The cursor will move to the top of the screen and EMACS will place a T every 8 spaces on the line under the ruler to show you where the tab stops are set.

Your screen should look like this:

```

      1      2      3      4      5      6      7      8
....5....0....5....0....5....0....5....0....5....0....5....0
█      T      T      T      T      T      T      T      T

EMACS 615 (.tab) *
Type a space, t, b, f, h, r, ?, or q

```

Notice the message under the mode line at the bottom of the screen:

Type a space, t, b, f, h, r, ?, or q

This message lists the command options you now have. They are explained in Table 8-1.

You cannot use the {CONTROL}, {ESCAPE}, or {RETURN} keys to move the cursor as long as the (.tab) notation appears on the status line. EMACS will recognize only the eight commands listed above until you type "q" to get back to your text. If you type some other command by mistake, EMACS will ignore it and print:

Unknown response

Table 8-1. Tab Command Options

<u>Command</u>	<u>Explanation</u>
{SPACE}	Moves cursor forward on the line under the ruler. Will delete a T.
t	Sets a tab stop at the column you specify.
f	Moves the cursor forward on the line under the ruler. Will not delete a T.
b	Moves the cursor backward on the line under the ruler. Will not delete a T.
h	Shifts the ruler left 60 spaces. Allows you to set tab stops in columns you normally don't see on the screen.
r	Shifts ruler back to normal position.
q	Signals EMACS that you want to quit setting tab stops and go back to your text.
?	Help. Prints a list of tab command options.

Let's try two of these commands now. First, type:

h

The ruler will shift left on the screen so that columns 60 to 140 are visible. Every eighth column will contain a T to show you where tab stops are set:

```

6          7          8          9          0          1          2          3
0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....
■ T          T          T          T          T          T          T          T

1 EMACS 615 (.tab) *
Type a space, t, b, f, h, r, ?, or q

```

These columns are not visible on a normal screen. Tab settings are available in these columns in case you have long tables or lines of text that extend beyond 80 characters. Chapter 10 explains how to view this extended text.





to delete the T. Now move back to the tab stop in column 16. Type:

b

as many times as it takes to move the cursor back to the T in column 16. Now type:

{SPACE}

The T will disappear. Your screen should now look like this:

```

      1         2         3         4         5         6         7         8
...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0
  T           T           T           T           T           T           T           T

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
Type a space, t, b, f, h, r, or q

```

You've set some tab stops. To go back to your text and try them out, type:

q

The ruler and tab stops will disappear from your screen. Your text will reappear with the cursor in its previous position.

Now type:

{CNTRL-I}

several times to move the cursor forward on any line. It will move to the tab stops you've set.

Setting Your Own Tab Stops

Now try setting tab stops again without specifying a default interval.  
Type:

```
{ESC}X settab {RETURN}
```

When you see the prompt:

```
Is there a default interval:
```

Type:

```
no{RETURN}
```

The ruler will again appear at the top of your screen with the cursor in column 1 and a T in column 80:

```

      1      2      3      4      5      6      7      8
....5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0
█

```

EMACS 615 (.tab) \*

Type a space, t, b, f, h, r, or q

EMACS is waiting for you to set the tab stops. Go ahead. Use any of the available (.tab) commands to set tab stops in columns 5, 15, 40, and 70. When you finish, your screen should look like this:

```

      1      2      3      4      5      6      7      8
....5....0....5....0....5....0....5....0....5....0....5....0....5....0
      T      T      T      T      T      T      T      T

```

EMACS 615 (.tab) \*

Type a space, t, b, f, h, r, or q

When you're sure that you've set your tab stops correctly, type:

q

You should now be back to your original place in the text. Use CONTROL-I to move the cursor forward on a line. It should move to the tab stops you've set.

### Setting Tabs From a Table

Sometimes you will not know in advance exactly where you want to place your tab stops. For example, you might be typing a table with column headings and not know the number of the column where each heading begins. The ESCAPE-X `Settabs_from_table` command will set the tab stops for you automatically. This command can be abbreviated to "ESCAPE X `setft`".

To practice this command, you will need to add some new lines to your file. First, type:

```
{ESC}>
```

to move the cursor to the end of your text. Make sure the cursor is on the last line.

```
{RETURN} ***** {CNTRL-X}{CNTRL-Z}S
```

```
{CNTRL-N} {RETURN}
```

You've cleared some space for new text. Now type:

```
COLUMN(A)
{ESC}5{SPACE}
COLUMN(B)
{ESC}5{SPACE}
COLUMN(C)
{ESC}5{SPACE}
COLUMN(D)
```

Finally, type:

```
{CNTRL-X}{CNTRL-Z}S
```

to center the columns on the screen. Your screen should look like this:

```

to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.
                                *****
■      COLUMN(A)      COLUMN(B)      COLUMN(C)      COLUMN(D)

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Place the cursor anywhere on the line you just typed. Now you're ready to set the tab stops. Type:

```
{ESC}X settabs_from_table{RETURN}
```

The cursor will move back to the beginning of the line. The prompt:

```
Tabs are reset
```

will appear under the mode line to tell you that your tab stops are automatically set at the beginning of each column. To try them out, use CONTROL-I to move to the beginning of each column.

Now move the cursor to the end of the current line. Type:

```
{ESC}2 {RETURN}
```

The cursor will move two lines below the current one to the left margin, creating two empty lines:

Now type the following commands and text without typing {RETURN}:

```
{CNTRL-I} 000 {CNTRL-I} 123 {CNTRL-I} 456 {CNTRL-I} 789
```

CONTROL-I now moves the cursor to the beginning of each column. Your screen should look like this:

```
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.
*****

COLUMN(A)    COLUMN(B)    COLUMN(C)    COLUMN(D)
000          123         456         789█

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

Moving Text to Tab Stops

The Control-X Control-I command inserts enough spaces before the cursor to move the cursor to the next tab stop, causing any following text to shift right. Try it. First, type:

```
{ESC}2 {RETURN}
```

to add two more empty lines to your text. Next, type:

```
321
```

Then move the cursor back to the beginning of the line so that the cursor is on the "3":

```

to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.
                                *****

      COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)
      000         123         456         789

█21

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Now type:

```
{CNTRL-X}{CNTRL-I}
```

Watch the number "321" move with the cursor to the first tab stop at "COLUMN(A)". Again, type:

```
{CNTRL-X}{CNTRL-I}
```

Watch the number "321" move with the cursor to the next tab stop:

```

to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.
                                *****

      COLUMN(A)      COLUMN(B)      COLUMN(C)      COLUMN(D)
      000            123            456            789
                                █21

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

### FILLING PARAGRAPHS

Often when you are editing text, the lines can become uneven after you've made additions or deletions. The ESCAPE-Q command will fill up these lines one paragraph at a time, so that your spacing is even. To try it, you'll need to add more text to the file.

First, use CONTROL-E to move the cursor to the end of the current line. Then type the following commands and text. There will be some typographical errors in the text. Go ahead and type them exactly the way you see them:

```

{ESC}3 {RETURN}
The COLUMNS {RETURN}
ABOVE{RETURN}
illustrate the use of the escape x setft command. escape X SETFT
{RETURN}
sets the tab stops to the beginning of each column {RETURN}
automatically. {ESC}2{RETURN}
These sentences illustrate the use fo some other editing commands.
{RETURN}
Character transpositions can be fixed with hte control t {RETURN}
command. Word transpositions be can fixed with{RETURN}
the T ESCAPE command.

```



Your screen should look like this:

```

                COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)
                000         123         456         789
                321

The COLUMNS
ABOVE
illustrate the use of the escape x setft command.  escape X SETFT
sets the tab stops to the beginning of each column
automatically.

These sentences illustrate the use fo some other editing commands.
Character transpositions can be fixed with hte control t
command.  Word transpositions be can fixed with
the T ESCAPE command.█

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Before you can use the ESCAPE Q command, you must tell EMACS that you want to fill paragraphs in the file. You do this by invoking fill mode. Do it now. Type:

```
{ESC}X Fill_on{RETURN}
```

The word "fill" should now be present on the mode line. It's there to remind you that fill mode is in effect. It will remain in effect until you tell EMACS you no longer want it available. You will do that further on in this chapter.

Now, you can see how paragraph filling works with the ESCAPE Q command. First, move the cursor to the "T" in the word "The" in the first paragraph of the new text that you've just typed. Then type:

```
{ESC}Q
```

Watch the lines fill up so that you have an even paragraph. The cursor will move to the end of the paragraph.

Your screen should look like this:

```

          COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)
          000         123         456         789
                    321

The COLUMNS ABOVE illustrate the use of the escape x setft command.
escape X SETFT sets the tab stops to the beginning of each column
automatically.

These sentences illustrate the use fo some other editing commands.
Character transpositions can be fixed with hte control t
command. Word transpositions be can fixed with
the T ESCAPE command.

EMACS 615 fill (practice) * <MISCEL>LILLI>PRACTICE

```

Sometimes when you insert more text onto a line that is almost full, part of the line will move off the screen. You can use ESCAPE Q to move this text back onto the screen again. Let's try that. Move the cursor to the "i" in the word "illustrate" in the next paragraph. Then add the following text to the sentence. Be sure to leave one space after the period and two spaces after the word "They". Type:

contain typographical errors. They

Part of your sentence will move off the screen to the right. Your screen should look like this:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the escape x setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some of the control characters. Character transpositions can be fixed with the control t command. Word transpositions can be fixed with the T ESCAPE command.

EMACS 615 fill (practice) \* <MISCEL>LILLI>PRACTICE

Now type:

{ESC}Q

Watch the lines of text fill up to form an even paragraph. The text that was shoved off the screen will move down to the next line.

Your screen should look like this:

```

          COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)
          000         123         456         789
                    321

The COLUMNS ABOVE illustrate the use of the escape x setft command.
escape X SETFT sets the tab stops to the beginning of each column
automatically.

These sentences contain typographical errors. They illustrate the
use fo some other editing commands. Character transpositions can be
fixed with hte control t command. Word transpositions be can fixed
with the T ESCAPE command.

EMACS 615 fill (practice) * <MISCEL>LILLI>PRACTICE

```

Fill mode is also helpful when you are creating new lines of text. If you want to avoid having text go off the screen, but you don't want to look at the screen while you're typing to decide when to press the RETURN key, you can let EMACS decide for you.

As long as fill mode is in effect, EMACS will automatically begin a new line for you as you type text on to new lines in the file. Keep in mind that this is different than inserting text onto a line that already exists. When you insert text onto an existing line, you must use ESCAPE Q, as you did in the last exercise.

For practice, try letting EMACS fill a paragraph as you type now. Move the cursor to a blank line in the file and type a few lines of text without typing {RETURN}. Notice how EMACS automatically begins a new line when the current one is filled up.

If you no longer want to have fill mode in effect, type:

```
{ESC}X Fill_off {RETURN}
```

The word "fill" will disappear from the mode line.

## COMMAND SUMMARY

In this chapter you've learned the following formatting commands:

<u>Command</u>	<u>Description</u>
{CNTRL-X}{CNTRL-Z}S	Centers Line
{ESC}I	Indents line the same as previous line
{ESC}Q	Fills a paragraph in fill mode
{ESC}X Fill_on	Tells EMACS you're planning to fill paragraphs by putting fill mode in effect
{ESC}X Fill_off	Turns off fill mode
{ESC}X Settab	Sets tabs stops
{ESC}X Setft	Sets tab stops from a table
{CNTRL-I}	Moves cursor to next tab stop
{CNTRL-X}{CNTRL-I}	Inserts spaces to next tab stop

## CHAPTER 9

## HELPFUL EDITING COMMANDS

The commands in this chapter are useful for editing text after it has been entered into a file. When you finish this chapter, you'll be able to:

- Move the cursor backward one sentence
- Move the cursor forward one sentence
- Move the cursor backward one paragraph
- Move the cursor forward one paragraph
- Fix transposed characters
- Fix transposed words
- Convert words to uppercase
- Convert words to lowercase

None of the commands in this chapter are essential when using EMACS. However, they will make your editing task much easier.

## MOVING THE CURSOR BACKWARD ONE SENTENCE

The ESCAPE A command moves the cursor backward one sentence. To try it, move the cursor to the end of the last sentence in your file. It should be resting on the period after the word "command". If you have any other practice text in the file after that sentence, delete it. Then type:

```
{ESC}A
```

The cursor will move back to the end of the previous sentence:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the escape x setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use fo some other editing commands. Character transpositions can be fixed with hte control t command. Word transpositions be can fixed with the T ESCAPE command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Type ESCAPE A a few more times and watch the cursor move back through your text sentence by sentence. This command is particularly helpful when you have long sentences that take up several lines on your screen.

#### MOVING THE CURSOR FORWARD ONE SENTENCE

The ESCAPE E command moves the cursor to the end of the current or following sentence. To try it, move the cursor to the "T" in the word "The" in the first paragraph. Then type:

{ESC}E

The cursor will move to the end of the current sentence:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the escape x setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use fo some other editing commands. Character transpositions can be fixed with hte control t command. Word transpositions be can fixed with the T ESCAPE command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Use the ESCAPE E command several more times to move the cursor forward through your text, sentence by sentence.

#### MOVING THE CURSOR BACKWARD ONE PARAGRAPH

The CONTROL-X [ command moves the cursor to the beginning of the current or previous paragraph. To try it, move the cursor to the "c" in the word "command" at the end of the last paragraph. Then type:

{CNTRL-X}[



The cursor will move to the blank line before the beginning of the paragraph:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the escape x setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use fo some other editing commands. Character transpositions can be fixed with hte control t command. Word transpositions be can fixed with the T ESCAPE command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Type this command one more time so that you move back through your text to the blank line above the preceding paragraph.

#### MOVING THE CURSOR FORWARD ONE PARAGRAPH

The CONTROL-X ] command moves the cursor to the space after the current paragraph. It works just like the CONTROL-X [ command. Try it now. Type:

{CNTRL-X}]

The cursor will move to the space following the current paragraph:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the escape x setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use fo some other editing commands. Character transpositions can be fixed with hte control t command. Word transpositions be can fixed with the T ESCAPE command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

#### FIXING TRANSPOSED CHARACTERS

A common editing task is fixing transpositions. The CONTROL-T command will fix transposed characters for you and save you some keystrokes. Try it now.

First, move the cursor to the word "fo". The two letters in this word are transposed. The word should really be "of". To fix this transposition, move the cursor to the space after the "o".

Your screen should look like this:

```

          COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)

          000         123         456         789

                   321

The COLUMNS ABOVE illustrate the use of the escape x setft command.
escape X SETFT sets the tab stops to the beginning of each column
automatically.

These sentences contain typographical errors. They illustrate the
use of some other editing commands. Character transpositions can be
fixed with the control t command. Word transpositions can be fixed
with the T ESCAPE command.

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Now type:

```
{CNTRL-T}
```

The two letters will change places to form the word "of".

Try this command again. Find the word "hte" in the same paragraph. It contains a transposition. Since the CONTROL-T command transposes the two letters preceding the cursor, place the cursor on the "e". Now type:

```
{CNTRL-T}
```

EMACS will transpose the "h" and the "t" to form the word "the":

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the escape x setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the control t command. Word transpositions be can fixed with the T ESCAPE command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

## FIXING TRANSPOSED WORDS

The ESCAPE T command fixes transposed words. To use it, position the cursor between the two words you want transposed. Try it now.

Find the word sequence "be can" in the second last sentence of your text. Place the cursor between the two words so that your screen looks like this:

```

          COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)
          000         123         456         789
                   321

The COLUMNS ABOVE illustrate the use of the escape x setft command.
escape X SETFT sets the tab stops to the beginning of each column
automatically.

These sentences contain typographical errors. They illustrate the
use of some other editing commands. Character transpositions can be
fixed with hte control t command. Word transpositions be can fixed
with the T ESCAPE command.

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Now type:

```
{ESC}T
```

The two words will exchange places so that they're in the correct order.

Try that again. Find the word sequence "T ESCAPE" farther along in the same sentence. Place the cursor between the two words. Now type:

```
{ESC}T
```

Watch the two words exchange places so that they're in the proper order:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the escape x setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the control t command. Word transpositions can be fixed with the ESCAPE **█** command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

## CONVERTING WORDS TO UPPERCASE

The ESCAPE U command converts the word following the cursor to uppercase and then moves the cursor to the end of the word. This command can save you many keystrokes.

To try it, move the cursor to the first occurrence of the word "escape". It is in the preceding paragraph. Place the cursor before the first "e".

Your screen should look like this:

```

          COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)
          000         123         456         789
                    321

The COLUMNS ABOVE illustrate the use of the escape x setft command.
escape X SETFT sets the tab stops to the beginning of each column
automatically.

These sentences contain typographical errors. They illustrate the
use of some other editing commands. Character transpositions can be
fixed with the control t command. Word transpositions can be fixed
with the ESCAPE T command.

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Now type:

```
{ESC}U
```

The word "escape" will be converted to uppercase and the cursor will move to the space after the word. Now type:

```
{ESC} U
```

Watch the "x" change to uppercase and the cursor move to the space after it:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the ESCAPE X setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the control t command. Word transpositions can be fixed with the ESCAPE T command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

If you want to convert more than one word to uppercase, you can use the ESCAPE <digit> prefix. To try it, move the cursor to the word sequence "control t" in the following paragraph. Place the cursor before the "c". Now type:

{ESC}2{ESC}U



Both words will change to uppercase and the cursor will move to the space after the "T":

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the ESCAPE X setft command. escape X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the CONTROL T command. Word transpositions can be fixed with the ESCAPE T command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Sometimes you will realize that a word should be in uppercase just after you've typed it. When this happens, you can use the ESCAPE <digit> command with a negative number to convert the word before the cursor to uppercase. Try it now. Move the cursor to the space after the word "escape" in the second sentence of the previous paragraph. Now type:

{ESC}-1{ESC}U

The word "escape" will convert to uppercase and the cursor will remain in its present position:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the ESCAPE X setft command. ESCAPE X SETFT sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the CONTROL T command. Word transpositions can be fixed with the ESCAPE T command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

#### CONVERTING WORDS TO LOWERCASE

The ESCAPE L command converts the word after the cursor to lowercase. It works just like the ESCAPE U command. Try it now. Move the cursor to the space before the word "SETFT". To change it to lowercase, type:

{ESC}L

"SETFT" will change to lowercase and the cursor will move to the space after it:

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The COLUMNS ABOVE illustrate the use of the ESCAPE X setft command. ESCAPE X setft sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the CONTROL T command. Word transpositions can be fixed with the ESCAPE T command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

You can use ESCAPE <digit> with ESCAPE L to convert more than one word to lowercase. To try it, move the cursor to the "C" in the word sequence "COLUMNS ABOVE". Now type:

{ESC}2{ESC}L

Both words will change to lowercase and the cursor will move to the space after "above":

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
000	123	456	789
	321		

The columns above illustrate the use of the ESCAPE X setft command. ESCAPE X setft sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the CONTROL T command. Word transpositions can be fixed with the ESCAPE T command.

EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

You can also use ESCAPE <digit> with a negative number to convert words preceding the cursor to lowercase.

#### COMMAND SUMMARY

You've learned the following commands in this chapter:

<u>Command</u>	<u>Description</u>
CONTROL-T	Transposes two characters
ESCAPE A	Moves cursor backward one sentence
ESCAPE E	Moves cursor forward one sentence
ESCAPE L	Converts word to lowercase
ESCAPE T	Transposes two words
ESCAPE U	Converts word to uppercase
CONTROL-X [	Moves cursor backward one paragraph
CONTROL-X ]	Moves cursor forward one paragraph

## CHAPTER 10

## OTHER USEFUL COMMANDS

This chapter describes three commands that may be useful to you in specific situations. Two of these commands change the way the text appears on your screen, making it easier to work with. The third command enables you to reach PRIMOS command level without leaving the EMACS editor.

The commands discussed include:

- Moving the screen horizontally
- Overlay mode
- Executing PRIMOS commands

## MOVING THE SCREEN HORIZONTALLY

Sometimes when you type text into a file, you may forget to press the RETURN key and your text will go off the screen. This text is not lost; you just can't see it because there is no room for it on the screen. Most terminal screens can only display 80 characters at a time.

There are two ways you can view this text that has disappeared. One way is to use the ESCAPE Q command. (See Chapter 9.) However, there may be times when you want your lines to be longer than 80 characters. In these cases you won't want to convert them to smaller lines, but you'll still want to see what you've typed. The ESCAPE X Set\_hscroll command permits you to do this by shifting your screen to the left horizontally until the text you want to see comes into view.

The ESCAPE X set\_hscroll command tells EMACS what column should be the leftmost column on the screen. After you type this command, EMACS will prompt you for a column number. EMACS will then shift the screen so that the column you've specified will appear in the leftmost column on the screen. Everything to the left of this column will move off the screen. The information to the right of it will come into view. When you want to get back to your original position in the text, you can type the ESCAPE X Reset command. It shifts the screen right, moving the leftmost column back to its original position.

Try these commands now. First, you'll need to add more text to your document. Move the cursor to the end of the last sentence in your file.

Then type:

```
{ESC}2 {RETURN}
```

You are ready to begin typing a new line of text. Go ahead and do that. Type the following sentences on one line. Be sure to leave two spaces after the first period. Do not type {RETURN}:

This is going to be a very long line. Part of this sentence is going to go off the screen.

Notice that when there is no more room on the screen for text, the cursor stops at the last column on the screen and the actual column number that the text has reached appears on the mode line. Your screen should look like this:

The columns above illustrate the use of the ESCAPE X setft command. ESCAPE X setft sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the CONTROL T command. Word transpositions can be fixed with the ESCAPE T command.

This is going to be a very long line. Part of this sentence is going to go off █

92 EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

The "92" appearing on the mode line tells you that the text off the screen extends to column 92. This means that there are 92 characters in your line, but you can only see the first 80 characters on the screen. There are 12 columns of text off the screen. If you shift the text left 12 columns, you will be able to see the missing text. Let's do that. Type:

```
{ESC}X set_hscroll
```

EMACS will prompt you with:

What is the horizontal column:

In response, type:

```
12
```

All the text on your screen will shift left 12 columns and the text that was off the screen will become visible on the right. The prompt:

```
hcol is 12
```

will appear under the mode line to remind you that you're looking at shifted text and that your leftmost column is now column 12. Your screen should look like this:

```

above illustrate the use of the ESCAPE x setft command.
tft sets the tab stops to the beginning of each column
ly.
```

```

nces contain typographical errors. They illustrate the
other editing commands. Character transpositions can be
the CONTROL T command. Word transpositions can be fixed
CAPE T command.
```

```
ng to be a very long line. Part of this sentence is going to go off the screen
```

```
92 EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE
```

```
hcol is 12
```

If you made any typos that you weren't able to see when you first entered this text, you can fix them now. When you're ready to get back to your original position, type:

```
{ESC}X reset
```

The text on your screen will shift right to its original position and the column number should disappear from the mode line.

## OVERLAY MODE

Text editors are commonly used for file updating. This often involves adding and deleting information to forms or tables. When you have a lot of information on a line but have only a small change to make, it is tedious to delete the old text before inserting the new information. Overlay mode changes the way text is inserted so that you can just type right over any unwanted information. When you insert new information in overlay mode, old information is not shoved over to the right as usual.

Let's try overlay mode now. Type:

```
{ESC}X overlay_on{RETURN}
```

The word "overlay" will appear in the mode line to remind you that overlay mode is in effect. The rest of your screen will not change.

Move the cursor to the first "0" in COLUMN(A). Now type:

```
abc
```

These characters will replace "000" without moving any other characters on the line. Now move the cursor to the "1" in COLUMN(B). Type:

```
def
```

Again, the new information will replace the old information without moving any other characters on the line. Now move the cursor to the "4" in COLUMN(C). Before you type more information into the file, type:

```
{ESC}X overlay_off{RETURN}
```

to get out of overlay mode.

The "overlay" prompt will disappear from the mode line. Overlay mode is no longer in effect. Now type:

```
ghi
```

This time, the other characters on the line will shift to the right as you add your new information.



Your screen should look like this:

```

to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.

```

\*\*\*\*\*

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
abc	def	ghi56	789
	321		

```

The columns above illustrate the use of the ESCAPE X setft command.
ESCAPE X setft sets the tab stops to the beginning of each column
EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Now you'll have to delete the old information. Use any method you like to do this. Notice how overlay mode saved you some keystrokes in the previous column.

Like fill mode, overlay mode is file specific. When you type the ESCAPE X overlay\_on command, it takes effect only in the file you are currently working with. If you want to put overlay mode in effect for all files you edit, you must issue this command each time you begin editing a new file.

#### EXECUTING PRIMOS COMMANDS

There may be times when you want to execute a PRIMOS command during the middle of an editing session. The CONTROL-X CONTROL-E command allows you to type a PRIMOS command, see its results, and then return to the file you're editing without leaving EMACS. This command can be a real time saver.

When executing PRIMOS commands in EMACS, keep in mind the following restrictions:

- You can execute only one PRIMOS command at a time.
- You cannot use any abbreviations for PRIMOS commands that you may have created. EMACS does not understand them.
- You cannot execute an interactive external command. (See the PRIMOS Commands Reference Guide, (FDR3108), to find out if these restrictions apply to a particular command.)

Try executing the LISTF (list contents of current directory) PRIMOS command now. Type:

```
{CNTRL-X}{CNTRL-E}
```

EMACS will respond with the prompt:

```
Primos command:
```

Now type:

```
listf{RETURN}
```

You will see some notations under the mode line telling you that your command is being executed. When execution is completed, your original text will disappear from the screen. EMACS will switch you to a new file which contains your output. In this case, your output will be a list of the files in your current directory. This list will be displayed on the screen. The mode line will read:

```
(file_output)
```

When you are ready to go back to your original file, type:

```
{CONTROL-X}{CONTROL-F}
```

When you see the prompt:

```
Find file:
```

Type:

```
practice
```

Your original text will return.

## COMMAND SUMMARY

In this chapter, you've learned the following commands:

<u>Command</u>	<u>Description</u>
{ESC}X set_hscroll	Makes specified column leftmost column
{ESC}X reset	Makes leftmost column first column on screen.
{ESC}X Fill_on	Puts fill mode in effect
{ESC}X Fill_off	Stops fill mode
{ESC}X Overlay_on	Puts overlay mode in effect
{ESC}X Overlay_off	Stops overlay mode
{CNTRL-X}{CNTRL-E}	Executes specified PRIMOS command

## CHAPTER 11

## BUFFERS AND WINDOWS

Up to this point, you've been working on one version of a single file. In EMACS, it is possible to create several different versions of a file, enabling you to choose the version you like best as your final copy. It is also possible to work on any number of different files during one editing session.

This chapter describes how to use multiple buffers and windows to edit multiple files and versions of files.

### BUFFERS

A buffer is a workspace that is set aside for you to use. EMACS uses buffers to organize your files. You can think of a buffer as a container and a file as the contents of the container. When you edit a file in EMACS, you are not actually looking at the file itself. You're looking at a copy of the file that EMACS has placed in a buffer.

Because EMACS always works with copies of files, it is possible to have many copies of the same file in different buffers and do different things to all of them.

### Mode Line Notations

You may have noticed that the filename "practice" always appears twice on the mode line at the bottom of your screen. It is written once in parentheses, and once at the end of its full PRIMOS pathname. The name in parentheses is name of the buffer you are currently using to edit your file. The PRIMOS pathname is there to tell you which file has been placed in the buffer, and the directory it came from. Until now, the buffer name and the filename have been the same. There will be times, however, when you will want these names to be different.

### Creating Buffers

So far, there has been no need for you to create a buffer. EMACS did it for you. Each time you used the CONTROL-X CONTROL-F command (See Chapter 4) to find your file at the beginning of an editing session, EMACS placed a copy of your file into a buffer automatically and gave the buffer the same name as your file.

Suppose, however, you want to make some changes to your current file but you also want to keep a copy of the file as it stands now on hand for reference? You would want to copy the current version into a different EMACS buffer before you made your changes. This could

present a problem since both the original version of the file and the changed version of the file would have the same name. EMACS does not allow you to have more than one buffer with the same name.

The CONTROL-X B and CONTROL-X CONTROL-R commands solve this problem. They allow you to create buffers that have different names from the files they contain.

CONTROL-X B creates or finds the buffer you specify. It lets you choose the buffer name, so that it is not automatically assigned the same name as the file it contains. If you choose the name of a buffer that already exists, EMACS will find that buffer for you and print its contents on the screen. If you choose a new name, EMACS will create an empty buffer with the name you specified. This buffer will not have a file name.

Try this command now. Type:

```
{CNTRL-X} B
```

The prompt:

Buffer:

will appear under the mode line. EMACS is now waiting for you to specify a buffer name. Let's create a new buffer called "oldfile." Type:

```
oldfile{RETURN}
```

Your text should disappear from the screen and the name "oldfile" should appear on the mode line in parentheses. The cursor should be in the upper-left corner of the screen. You've created a buffer called "oldfile".

Do not worry about saving the text in your original buffer. EMACS will keep the buffer exactly the way you left it until you exit EMACS. You will go back to the original buffer later on in this chapter. If you must exit EMACS before you finish this chapter, see the explanation of the CONTROL-X CONTROL-W command further on.

Keep in mind that a buffer is only a workspace that is used to hold text. So far, all you have is an empty buffer. It is really of no use to you unless there is text in it.

There are two ways to put text into a buffer. One way is just to type text right in. The other way is to use the CONTROL-X CONTROL-R command to insert an existing file into the buffer. Since we are eventually going to compare two versions of the same file, we'll use CONTROL-X CONTROL-R. Try it now. Type:

```
{CNTRL-X}{CNTRL-R}
```

The prompt:

Read File:

will appear under the mode line. EMACS is waiting for you to supply a filename. Type:

practice{RETURN}

A copy of the file "practice" will appear on your screen. The full pathname of this file will appear on the mode line to remind you that the file "practice" is in your buffer even though the name of your buffer is "oldfile".

Your screen should look like this:

```

      ■
                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
      which I would make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, would produce chickens,
and by and by I shall have quite a large poultry-yard.
      *****

And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting action to the word,
EMACS 615 (oldfile) <MISCEL>LILLI>PRACTICE

File read: <MISCEL>LILLI>PRACTICE

```

#### USING MULTIPLE BUFFERS

You now have two copies of the file "practice" residing in two separate buffers in EMACS. You are looking at the buffer "oldfile". We'll use this buffer to hold the file "practice" as it stands now, without any further alterations. We'll use the original buffer, "practice", to make more changes to its copy of the file "practice". Let's go back to that buffer now. Type:

{CNTRL-X} B

When you see the prompt:

Buffer:

Type:

```
practice{RETURN}
```

Your original buffer, "practice", will return to your screen, exactly as you left it. EMACS has not only remembered how you left your text, it has remembered how you left your screen as well. The cursor will be exactly where it was when you left the buffer. Note that the buffer notation in parentheses on the mode line now reads "practice". Your screen should look like this:

```
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.
```

\*\*\*\*\*

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
abc	def	ghi	789
	321		

The columns above illustrate the use of the ESCAPE X setft command.  
EMACS 615 (practice) \* <MISCEL>LILILI>PRACTICE

Now make some changes to your text. Use any method you like to position the cursor on the line after the second row of stars (\*):

```

to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.

```

\*\*\*\*\*

■

COLUMN(A)	COLUMN(B)	COLUMN(C)	COLUMN(D)
abc	def	ghi	789
	321		

The columns above illustrate the use of the ESCAPE X setft command.  
EMACS 615 (practice) \* <MISCEL>LILLI>PRACTICE

Next type:

```
{CNTRL-@}
```

to set the mark on the line. (See Chapter 5 if you've forgotten how this command works and what it does.) Now type:

```
{ESC}>
```

to move the cursor to the end of the file. You've defined a region.

Now type:

```
{CNTRL-W}
```



The region you've just defined will disappear and your cursor will return to to the place where you set the mark:

```

to say to them."
Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.

```

```

*****

```

```

EMACS 615 (practice) * <MISCEL>LILLI>PRACTICE

```

Now let's create one more buffer. Type:

```
{CMTL-X} B
```

When you see the prompt:

```
Buffer:
```

Type:

```
columns{RETURN}
```

The buffer "practice" will disappear and a new, empty buffer called "columns" will take its place.

This new buffer needs some text. Type:

```
{CMTL-Y}
```

The lines that you just deleted in the previous buffer will reappear in your current buffer:

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the CONTROL T command. Word transpositions can be fixed with the ESCAPE T command.

This is going to be a very long line. Part of this sentence is going to go off

█

EMACS 615 (columns) \*

You are looking at the end of the lines you previously deleted. To see the rest of the lines, type:

{ESC}V

Even though you killed this text while you were in another buffer, EMACS remembered it. You can use CONTROL-Y to "yank" back your most recently killed text in any buffer.

#### GETTING A LIST OF BUFFERS

So far, you've worked with three different buffers in one editing session. If you wanted to, you could create more. As the number of buffers you work with increases, you may forget which buffers contain what files. The CONTROL-X CONTROL-B command takes care of this problem. It prints a list of buffers and the files they contain at the top of your screen. Try it now. Type:

{CNTRL-X}{CNTRL-B}

Part of your text will disappear. A double line of dashes (—) will appear near the top of your screen. Two columns of information will appear above the dashes.

Your screen should look similar to this:

```

oldfile *          50<MISCEL>LILLI>PRACTICE
practice *        50<MISCEL>LILLI>PRACTICE
file_output      13
.start_up        3<MISCEL>EMACS_LIB>EMACS_STARTUP
=====
█ 321

```

The columns above illustrate the use of the ESCAPE X setft command. ESCAPE X setft sets the tab stops to the beginning of each column automatically.

These sentences contain typographical errors. They illustrate the use of some other editing commands. Character transpositions can be fixed with the CONTROL T command. Word transpositions can be fixed with the ESCAPE T command.

This is going to be a very long line. Part of this sentence is going to

EMACS 615 (columns) \*

The left column above the dashes lists all the buffers you used during your editing session, excluding the buffer you are currently working with. The buffers are listed in order of creation. Those buffers created first are at the bottom of the list. Those buffers created more recently are at the top. You may see unfamiliar buffer names preceded by a period in this list. These buffers are used by EMACS only. They may not have corresponding pathnames.

The right column above the dashes lists the pathnames of the files corresponding to the buffers in the left column. A number in front of a pathname indicates the number of lines in the file.

To remove the list of buffers from the top of your screen, type:

```
{CNTRL-L}
```

The list of buffers will disappear and the text it displaced will reappear on the screen. The list of buffers will also disappear from your screen when you make the next change to your text.

## SAVING TEXT IN BUFFERS

To save text permanently, you must store it in a PRIMOS file. So far, you've been using the CONTROL-X CONTROL-S command to do this. CONTROL-X CONTROL-S automatically stores your text in the file listed on the mode line. Sometimes, you will want to store the text under a different filename. You can do this with the CONTROL-X CONTROL-W command.

CONTROL-X CONTROL-W works like CONTROL-X CONTROL-S, but it prompts you for a filename. This command is useful when you have different versions of the same file in different buffers. It is also useful when you've created a buffer and added text to it that did not come from a file, as is the case with the current buffer, "columns". Try saving the text in this buffer now.

First, try using CONTROL-X CONTROL-S. Type:

```
{CNTRL-X}{CNTRL-S}
```

EMACS will respond with the prompt:

```
No default file name for this buffer
```

This command does not work. Now type:

```
{CNTRL-X} {CNTRL-W}
```

This time, EMACS will respond with the prompt:

```
Write file:
```

EMACS is waiting for you to provide a filename. In this case, it is convenient to use the name of the buffer as the filename, since it will help you remember what is in the file. Type:

```
columns{return}
```

The text in the buffer "columns" is now permanently stored in the file "columns". The "File written" notation will appear below it to let you know that your file has been saved.

You've saved the text in one buffer. There are two more to go. Let's save the text in the buffer "practice" next. Type:

```
{CNTRL-X} B
```

When you are prompted for a buffer name, type:

```
practice{RETURN}
```

The buffer "practice", containing the changed version of the file "practice", will appear on your screen. Before you save this text in a permanent file, you must choose a filename. Although you have two versions of the file "practice", you can't have two PRIMOS files with that name. Let's call this altered version "practice.2".

Now you're ready to save the text. Type:

```
{CNTRL-X} {CNTRL-W}
```

When you see the prompt

```
Write File:
```

```
type:
```

```
practice.2{RETURN}
```

The usual "File written" prompt will appear at the bottom of your screen to tell you that your file was saved. You have one more file left to do. Type:

```
{CNTRL-X} B
```

When prompted for a buffer name, type:

```
oldfile{RETURN}
```

The buffer "oldfile", containing the unaltered version of the file "practice", will appear on your screen. You need a name for this version of the file "practice". Let's call it "practice.1".

Now go ahead and save this file. Type:

```
{CNTRL-X}{CNTRL-W}
```

When prompted for a filename, type:

```
practice.1{return}
```

The "File written" prompt will appear under the mode line. You've saved your text.

### Buffers as Working Environments

Because EMACS remembers the state of each buffer, different buffers can be set up to perform specialized editing tasks. For example, suppose your work for the day included updating several files containing tables and typing a manuscript into a new file. When you updated the tables, it would be helpful to have overlay mode in effect. While you typed the manuscript, fill mode would be useful. You could create two different buffers, one with fill mode in effect, and one with overlay mode in effect. When you had a table to edit, you might do this:

- Use CONTROL-X B to create or find your overlay buffer.
- Use ESCAPE X Overlay\_on to put overlay mode in effect.
- Use CONTROL-X CONTROL-R to place your file containing the table into the buffer.
- Make your changes to the table.
- Use CONTROL-X CONTROL-W to make your changes permanent.
- Use CONTROL-X CONTROL-R to place a new file with tables into your buffer and repeat the above process.

When you were ready to type your manuscript, you could create a different buffer and use ESCAPE X fill\_on to put fill mode in effect.

You can change buffers as many times as you like. EMACS will remember the changes you made to each buffer and keep them in effect for the duration of your editing session.

### Chapter Summary

So far in this chapter you have:

- Learned that buffers are "containers" used by EMACS to keep track of the text that you edit.
- Used the CONTROL-X B command to create and name buffers.
- Used the CONTROL-X CONTROL-R command to put a file into a buffer.
- Used the CONTROL-X CONTROL-W command to save text permanently in a PRIMOS file.
- Used the CONTROL-X CONTROL-B command to get a list of the buffers used during the current editing session.

## MULTIPLE WINDOWS

Up to this point, you've been able to view only one buffer at a time on your screen. Sometimes, however, it will be useful for you to have text from two different buffers visible at once. For instance, in the first part of this chapter, you created two different versions of the file "practice". If you wanted to compare them to see how they differ, it would be useful to see parts of both of them displayed on the screen. EMACS' multiple window feature makes this possible.

The CONTROL-X 2 command divides your screen into two windows. Each window contains a different buffer. Try this command now. Type:

{CTRL-X} 2

A line of dashes (—) will appear across your screen, dividing it in half. The text that previously occupied the entire screen will occupy only the top half. This top half is called window one. The cursor will move to the upper left corner of the bottom half of your screen. This bottom half is called window two. It will be empty. The buffer notation in parentheses on the mode line will read "alternate". Your screen should look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out ot milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
-----
█

EMACS 615 (alternate)

```

Both windows contain buffers. However, you can only work with text in one buffer at a time. The name of the buffer you are currently working with is always displayed on the mode line. In this case, the name of the current buffer is "alternate". EMACS assigns this name to the buffer in window two until you specify otherwise.

You are now free to type any EMACS commands you like. Since the cursor is in window two, any command you type will work only in this window. If you try to move the cursor above the dotted line, it won't work. There is a special command to do this. You will use it later on in this chapter.

The first thing you want to do is put a file in your empty buffer. Since the file "practice.1" is in window one, let's use CONTROL-X CONTROL-F to place the file "practice.2" in window two. Type:

```
{CNTRL-X} {CNTRL-F}
```

When prompted for a filename, type:

```
practice.2{RETURN}
```

Since CONTROL-X CONTROL-F automatically places the file you specify in a buffer with the same name, the buffer name will change from "alternate" to "practice.2". If you had used CONTROL-X CONTROL-R to place your file in the buffer, the name of the buffer would have remained "alternate" until you specified otherwise. Your screen should look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out ot milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
-----
■
                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out ot milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
EMACS 615 (practice.2) <MISCEL>LILLI>PRACTICE.2

```



You're now looking at two different versions of the file "practice". You're ready to compare them. Remember that because the cursor is in window two, any commands you type will apply to the text in window two only. Now type:

```
{CNTRL-V}
```

several times to move forward through your file in window two. Notice how the text in window one remains unchanged. Now type:

```
{ESC}>
```

to move the cursor to the end of the file:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out ot milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
-----
in the air in a moment!
Moral: Do not count your chickens
before they are hatched.

                                *****

■

EMACS 615 (practice.2) <MISCEL>LILLI>PRACTICE.2

```

Now let's do the same thing in window one. First, we'll have to move the cursor to window one. The CONTROL-X O command will do this. It means "move to the other window." Try it. (Be sure to type the letter "O" and not the digit zero.) Type:

```
{CNTRL-X} O
```

The cursor will move back to where it was in your original file before you created two windows. The buffer name and pathname will change to reflect the text in window one:

```

■          THE MILKMAID AND HER PAIL
           from Aesop's Fables

A farmer's daughter had been out ot milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
-----
in the air in a moment!
Moral: Do not count your chickens
before they are hatched.

          *****

EMACS 615 (practice.1) <MISCEL>LILLI>PRACTICE.1

```

You are now working with the file named "practice.1" in the buffer named "practice.1". Any commands you type will work in window one only. First, move the cursor to the end of this file. Type:

```
{ESC}>
```

The end of the file "practice.1" will appear in window one. If we compare this text with the text in window two, you can see that the two files are different. Let's find the place where they begin to differ. Type:

```
{ESC}V
```

as many times as it takes to make the text that is now in window two appear in window one also.

Your screen should look similar to this:

```

she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.
*****

          COLUMN(A)    COLUMN(B)    COLUMN(C)    COLUMN(D)
-----
in the air in a moment!
Moral: Do not count your chickens
before they are hatched.
*****

EMACS 615 (practice.1) <MISCEL>LILLI>PRACTICE.1

```

You can see that the two files contain some of the same information. You can also see that "practice.1" contains information that is not present in "practice.2".

Now go back to window 2. Type:

```
{CNTRL-X} O
```

The cursor will move down to its previous position in window two.

Now suppose you want to move back and forth through your file in window one for reference purposes, but you don't plan to make any editing changes in that file. Using the commands you've learned so far, you would have to type CONTROL-X O to move the cursor to window two, type ESCAPE V or CONTROL-V, and then type CONTROL-X O again to get back to window two. This could become tedious after a while. There are two commands that will solve this problem.

CONTROL-X V means "move the other window backward one screen." It will move the text in window one backward one screen without changing buffers. The cursor will remain in your current window. Try it now. Type:

```
{CNTRL-X}V
```

As the text in window one moves backward, the cursor will move to window one very quickly and then return to its place in window two. Your screen should look like this:

and by and by I shall have quite a large poultry-yard.

\*\*\*\*\*

And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."

Forgetting all about the pail,  
and suiting the action to the word,  
she tossed her head.  
Down went the pail, all the milk

---

in the air in a moment!  
Moral: Do not count your chickens  
before they are hatched.

\*\*\*\*\*

EMACS 615 (practice.2) <MISCEL>LILLI>PRACTICE.2

ESCAPE CONTROL-V means "move the other window forward one screen." It works just like CONTROL-X V except that it moves the text forward. Try it. Type:

{ESC}{CNTRL-V}

The text in window one will move forward to its previous position. Your screen should look like this:

```

Forgetting all about the pail,
and suiting the action to the word,
she tossed her head.
Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.          *****

-----
in the air in a moment!
Moral: Do not count your chickens
before they are hatched.

          *****
■

EMACS 615 (practice.1) <MISCEL>LILLI>PRACTICE.1

```

You've made some changes to the text in window two. Now you're ready to make them permanent. You can use either CONTROL-X CONTROL-W or CONTROL-X CONTROL-S. Since there is no need to specify a different filename, we'll use CONTROL-X CONTROL-S. Type:

```
{CNTRL-X}{CNTRL-S}
```

You've saved the text in window two. Now move back to window one and do the same thing. Type:

```
{CNTRL-X} O
```

to move back to window one. Now type:

```
{CNTRL-X}{CNTRL-S}
```

to save your text.

So far you've divided your screen into two windows and used the windows to view two different files at the same time. You've moved the cursor back and forth between the windows and edited each file. Now you're ready to get back to only one window again.

The CONTROL-X 1 returns your screen to its normal state. The buffer in the current window will occupy the entire screen. The buffer in the other window will disappear. Let's try this command now. Type:

```
{CNTRL-X}1
```

The file "practice.1" should now occupy your entire screen. The cursor position should be the same. If the cursor had been in window two when you typed CONTROL-X 1, the file in window two would have become the current file and window one would have disappeared.

The file that disappeared with window two is not lost. If you need to look at it again, you can use CONTROL-X CONTROL-F, CONTROL-X B, or CONTROL-X 2.

#### COMMAND SUMMARY

In this chapter, you've learned about the following commands.

<u>Command</u>	<u>Description</u>
{CNTRL-X}B	Finds buffer created during current editing session. Creates new buffer if one specified is not found.
{CNTRL-X} {CNTRL-F}	Finds existing file and places it in buffer with same name. Will create new file with specified name if one does not exist.
{CNTRL-X} {CNTRL-R}	Finds existing file and places it in current buffer. Buffer does not necessarily have same name as file. If file does not exist, a new one is not created. Will prompt for deletion of buffer contents if buffer is full.
{CNTRL-X} {CNTRL-S}	Saves editing changes and writes them to the filename listed on the mode line.
{CNTRL-X} {CNTRL-W}	Saves editing changes but prompts for a filename to use. Filename does not have to be name listed on mode line.
{CNTRL-X} 2	Divides screen into two horizontal windows.
{CNTRL-X} 1	Starts showing only one window again.
{CNTRL-X} 0	Moves cursor to other window.
{CNTRL-X} V	Moves other window backward one screen.
{ESC}{CNTRL-V}	Moves other window forward one screen.

## CHAPTER 12

## KEYBOARD MACROS

When you are editing text in a file, you will often find yourself typing the same sequence of commands over and over again. For example, suppose you want to double-space a block of text that is currently single-spaced. To do this, you might move the cursor to the end of a line, insert a blank line after it, and then move to the next nonblank line to repeat the process. While these operations are simple, it would save time if you could combine this group of commands into a single command. You can do this by defining a keyboard macro.

A keyboard macro is a group of commands that EMACS saves together so that you can execute them later with one command. You define the keyboard macro by telling EMACS which commands you want saved.

## DEFINING A KEYBOARD MACRO

Defining a macro is very simple. All you have to do is type the "CONTROL-X (" command to signal EMACS that you want the following commands you type to be saved as a keyboard macro. Follow {CNTRL-X}( with the sequence of commands you want saved. Finally, type the "CONTROL-X )" command to tell EMACS that you've stopped defining your macro.

As you define your macro, the commands you type will be executed so you can see the effects they are having on your text. If you make a mistake in the middle of your definition, you can stop right away by typing CONTROL-X ) to end the macro definition. You can then start over again with the CONTROL-X ( command.

Let's define a simple keyboard macro now. Suppose you want to double-space your current file from the first line to the second line of stars(\*). You will have to repeat the following sequence of commands once for each line that you want to convert:

- Type CONTROL-E to move the cursor to the end of a line.
- Type CONTROL-O to insert a blank line after the current one.
- Type ESCAPE 2 CONTROL-N to move the cursor down to the next nonblank line.

Since this same group of commands must be typed over and over again, it will be easier to save them as one keyboard macro. Do that now.

First, use any method you like to move the cursor anywhere in the first sentence of the file. Then type:

```
{CNTRL-X}(
```

to signal EMACS that you're about to define a macro. The notation:

```
{Macro}
```

will appear on the mode line to remind you that any commands you type will be considered part of the macro definition. Now type:

```
{CNTRL-E}
```

your cursor will move to the end of the current line:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
      which I would make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, would produce chickens,
and by and by I shall have quite a large poultry-yard.
      *****

And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
and suiting the action to the word,
EMACS 615 {Macro} (oldfile) <MISCEL>LILLI>PRACTICE.1

```

Next, type:

```
{CNTRL-O}
```



A blank line will appear under the current line:

THE MILKMAID AND HER PAIL  
from Aesop's Fables

A farmer's daughter had been out to milk

the cows, and was returning to the dairy  
carrying her pail of milk upon her head.  
As she walked along, she began to daydream  
in the following manner:

"The milk in this pail will provide me with cream,  
which I would make into butter and take to market  
to sell. With the money I will buy a number of eggs,  
and these, when hatched, would produce chickens,  
and by and by I shall have quite a large poultry-yard.

\*\*\*\*\*

And all the young fellows will admire it,  
and come and make love to me,  
but I shall toss my head and have nothing  
to say to them."

Forgetting all about the pail,

EMACS 615 {Macro} (oldfile) <MISCEL>LILLI>PRACTICE.1

Now type:

{ESC}2{CNTRL-N}

This will move the cursor down to the next nonblank line:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
      which I would make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, would produce chickens,
and by and by I shall have quite a large poultry-yard.
      *****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
Forgetting all about the pail,
EMACS 615 {Macro} (oldfile) <MISCEL>LILLI>PRACTICE.1

```

Finally, type:

```
{CNTRL-X} )
```

to signal EMACS that you've finished your macro definition. Notice that the {macro} notation disappears from the mode line. You've defined a macro.

#### EXECUTING A MACRO

You can now execute all the commands in your macro with one command. To do this, use the CONTROL-X E command. This command executes the most recently defined keyboard macro. Try it now. Type:

```
{CNTRL-X} E
```

A blank line will appear under the current line and the cursor will move down to the next nonblank line. Your screen should look like this:

```

                                THE MILKMAID AND HER PAIL
                                from Aesop's Fables

A farmer's daughter had been out to milk
the cows, and was returning to the dairy
carrying her pail of milk upon her head.
As she walked along, she began to daydream
in the following manner:
    "The milk in this pail will provide me with cream,
      which I would make into butter and take to market
to sell. With the money I will buy a number of eggs,
and these, when hatched, would produce chickens,
and by and by I shall have quite a large poultry-yard.
      *****
And all the young fellows will admire it,
and come and make love to me,
but I shall toss my head and have nothing
to say to them."
EMACS 615 (oldfile) <MISCEL>LILLI>PRACTICE.1

```

So far, you've inserted two blank lines. You still have many lines to do. Use the ESCAPE <digit> command to repeat your macro enough times to double-space the rest of the file. There are 21 single-spaced lines left. That means you must execute your macro 21 more times. Go ahead and do it. Type:

```
{ESC}21{CNTRL-X} E
```

Your entire file up to the second row of stars should now be double-spaced:

```

and suiting the action to the word,
she tossed her head.

Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!

Moral: Do not count your chickens
before they are hatched.

```

```

*****

```

```

          COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)
          abc         def         ghi         789

EMACS 615 (oldfile) * <MISCEL>LILLI>PRACTICE.1

```

#### DEFINING A SECOND MACRO

You can define as many macros as you like during one editing session. However, the CONTROL-X E command will execute only the most recently defined macro. Since you've already double-spaced your file, you don't need your current macro. You're free to define a new one. Let's do that now.

This time, we'll try something a little more complicated. Suppose that you have a column of numbers 20 lines long and you want to shorten it into two 10 line columns, both beginning on the same line.

After placing the cursor where you want the second column to begin, you will have to follow this command sequence:

- Type CONTROL-@ to mark the cursor position.
- Type CONTROL-A to go back to the beginning of the line. (This moves the cursor back to column one.)
- Type ESCAPE 10 CONTROL-N to move the cursor down 11 lines. (Line 11 begins the second half of the column. It will become the first line of the new column.)
- Type CONTROL-K to kill the line.
- Type CONTROL-X CONTROL-X to move the cursor back to the mark for column two.
- Type CONTROL-Y to yank back the the line you killed. (This moves the line that was formerly in column one to column two.)
- Type CONTROL-N to move the cursor to the next line.
- Type CONTROL-X CONTROL-I to move the cursor to the beginning of the second column.

This entire sequence of commands must be executed 10 times to create the two columns. If you create a macro to execute these commands, you only have to type them once.

To practice this macro, you'll first need to create a column that is 20 lines long. Do this now. Place the cursor on the "a" under COLUMN(A). Then type:

```
{CNTRL-K}
```

to delete the entire line. Move the cursor to the line containing the "321" and type CONTROL-K again to delete that line. Now move the cursor back to the "C" in "COLUMN(A)". Type:

```
{ESC}2{CNTRL-N}
```

to move the cursor down two lines. You're ready to create your column.

Type the numbers 1 to 20 in COLUMN(A) so that each one is on a separate line.

Your screen should look like this:

```

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20■

```

The columns above illustrate the use of the ESCAPE X setft command. ESCAPE X setft sets the tab stops to the beginning of each column automatically.  
EMACS 615 (oldfile) \* <MISCEL>LILLI>PRACTICE.1

You now have one 20 line column. To convert it into two 10 line columns, you'll need to execute the sequence of commands outlined above. You'll execute them once for practice, and then convert them into a macro. Before you do that, however, you must set tab stops at the beginning of each column. Do that now. Move the cursor to the word "COLUMN(A)". Then type:

```
{ESC}X setft{RETURN}
```

The prompt:

```
Tabs have been reset
```

will appear under the mode line.

Next move the cursor to the line containing the "1". Then place it under the "C" in "COLUMN(B)".

Your screen should look like this:

```

Down went the pail, all the milk
was spilled, and all her fine castles
in the air vanished in a moment!
Moral: Do not count your chickens
before they are hatched.

                *****

                COLUMN(A)   COLUMN(B)   COLUMN(C)   COLUMN(D)
                1           █
                2
                3
                4
                5
                6
EMACS 615 (oldfile) * <MISCEL>LILLI>PRACTICE.1

```

Now you're ready to define your macro. First, type:

```
{CNTRL-X}(
```

to tell EMACS that you're going to define a macro. The word "macro" will appear on the mode line. Now type:

```
{CNTRL-@}
```

to mark the place where you want the second column to begin. There will be no change to your screen. Then type:

```
{CNTRL-A}
```

to move the cursor to the beginning of the line. Next, type:

```
{ESC}10{CNTRL-N}
```

to move the cursor down to the beginning of line 11. The text on this line and the ones following it will become the second column.

Your screen should look like this:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
█ 11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
  
EMACS 615 {Macro} (oldfile) * <MISCEL>LILLI>PRACTICE.1
```

Then type:

```
{CONTROL-K}
```

to kill the text on the line. A blank line will remain.



Your screen should look like this:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
█  
12  
13  
14  
15  
16  
17  
18  
19  
20  
  
EMACS 615 {Macro} (oldfile) * <MISCEL>LILLI>PRACTICE.1
```

Now type:

```
{CONTROL-X}{CONTROL-X}
```

The cursor will move up to COLUMN(B) where you set the mark. Your screen should look like this:

```
1      ■
2
3
4
5
6
7
8
9
10
12
13
14
15
16
17
18
18
20

EMACS 615 {Macro} (oldfile) * <MISCEL>LILLI>PRACTICE.1
```

Next, type:

```
{CONTROL-Y}
```

to yank back the text that you killed in COLUMN(A). Your screen should look like this:

```

1          11
2
3
4
5
6
7
8
9
10
12
13
14
15
16
17
18
19
20

EMACS 615 {Macro} (oldfile) * <MISCEL>LILLI>PRACTICE.1
```

Then, type:

```
{CNTRL-N}
```

to move the cursor to the next line. Next, type:

```
{CNTRL-X}{CNTRL-I}
```

to move the cursor to the beginning of COLUMN(B). To tell EMACS that you've stopped defining your macro, type:

```
{CNTRL-X})
```

The word "macro" will disappear from the mode line. Your screen should look like this:

```
1      11
2      █
4
5
6
7
8
9
10
12
13
14
15
16
17
18
18
20

EMACS 615 (oldfile) * <MISCEL>LILLI>PRACTICE.1
```

You've executed the command sequence to change one column into two columns once. You still need to execute this sequence nine more times. Do that now. Type:

```
{ESC}9{CNTRL-X}E
```

You should now have two 10 line columns on your screen:

```

1          11
2          12
3          13
4          14
5          15
6          16
7          17
8          18
9          19
10         20█

EMACS 615 (oldfile) * <MISCEL>LILLI>PRACTICE.1

```

In this chapter, you've learned how to create and execute simple keyboard macros. If you would like to learn more about keyboard macros, see the EMACS Reference Guide. It describes how to execute keyboard macros with different keystrokes as well as how to save them for future use.

#### COMMAND SUMMARY

<u>Command</u>	<u>Description</u>
{CNTRL-X} (	Begins keyboard macro definition
{CNTRL-X} )	Ends keyboard macro definition
{CNTRL-X} E	Executes most recent keyboard macro

## APPENDIX A

## BASIC COMMAND SUMMARY

The following is a list of the basic EMACS commands described in this book. The commands are sorted by function and each one is followed by the chapter in this book where it was first introduced. For a list of all EMACS commands, see Appendix B or Appendix C.

Character Cursor Commands

CONTROL-B	Chapter 3
CONTROL-F	Chapter 3

Word Cursor Commands

FESCAPE B	Chapter 3
ESCAPE F	Chapter 3

Line Cursor Commands

CONTROL-A	Chapter 4
CONTROL-E	Chapter 4
CONTROL-N	Chapter 3
CONTROL-Z	Chapter 3

Screen Cursor Commands

CONTROL-V	Chapter 4
ESCAPE V	Chapter 4
ESCAPE <	Chapter 4
ESCAPE >	Chapter 4

Search Commands

CONTROL-R	Chapter 4
CONTROL-S	Chapter 4

Character Delete Commands

CONTROL-D	Chapter 3
CONTROL-{BACKSPACE}	Chapter 3

Word Delete Commands

ESCAPE D	Chapter 3
ESCAPE {BACKSPACE}	Chapter 3
CONTROL-K	Chapter 5

Line and Region Delete Commands

CONTROL-K	Chapter 5
-----------	-----------

CONTROL-W Chapter 5

Move Text Commands

CONTROL-K Chapter 5  
 CONTROL-O Chapter 4  
 CONTROL-W Chapter 5  
 CONTROL-Y Chapter 5  
 CONTROL-X CONTROL-X Chapter 5  
 CONTROL-@ Chapter 5

Create, Name, and Save File Commands

CONTROL-X CONTROL-F Chapter 4  
 CONTROL-X CONTROL-S Chapter 3  
 CONTROL-X CONTROL-W Chapter 3

Exit Commands

CONTROL-P Chapter 6  
 CONTROL-X CONTROL-C Chapter 3

Help Commands

CONTROL-\_ Chapter 6  
 CONTROL-\_A Chapter 6  
 CONTROL-\_C Chapter 6  
 CONTROL-\_L Chapter 6

Execution Control Commands

CONTROL-C Chapter 4  
 CONTROL-G Chapter 6  
 ESCAPE <digit > Chapter 3

Replacement Commands

ESCAPE X Replace Chapter 7  
 ESCAPE % Chapter 7

Formatting Commands

CONTROL-I Chapter 8  
 CONTROL-X CONTROL-I Chapter 8  
 CONTROL-X CONTROL-Z S Chapter 8  
 ESCAPE I Chapter 8  
 ESCAPE Q Chapter 8  
 ESCAPE X Fill\_on Chapter 8  
 ESCAPE X Fill\_off Chapter 8  
 ESCAPE X Settab Chapter 8  
 ESCAPE X Setft Chapter 8  
 ESCAPE X Overlay\_on Chapter 10  
 ESCAPE X Overlay\_off Chapter 10

Sentence Movement Commands

ESCAPE A	Chapter 9
ESCAPE E	Chapter 9

Paragraph Movement Commands

CONTROL-X [	Chapter 9
CONTROL-X ]	Chapter 9

Case Conversion Commands

ESCAPE L	Chapter 9
ESCAPE U	Chapter 9

Transposition Commands

CONTROL-T	Chapter 9
ESCAPE T	Chapter 9

Buffer Commands

CONTROL-X 1	Chapter 11
CONTROL-X 2	Chapter 11
CONTROL-X 0	Chapter 11
CONTROL-X V	Chapter 11
ESCAPE CONTROL-V	Chapter 11

Buffer Commands

CONTROL-X B	Chapter 11
CONTROL-X CONTROL-F	Chapter 11
CONTROL-X CONTROL-R	Chapter 11
CONTROL-X CONTROL-S	Chapter 11
CONTROL-X CONTROL-W	Chapter 11

Commands for Viewing Text off the Screen

ESCAPE X Set_hscroll	Chapter 10
ESCAPE X Reset	Chapter 10

Commands for Keyboard Macros

CONTROL-X (	Chapter 12
CONTROL-X )	Chapter 12
CONTROL-X E	Chapter 12

Executing a PRIMOS Command

CONTROL-X CONTROL-E	Chapter 12
---------------------	------------



## APPENDIX B

## COMPLETE ALPHABETICAL SUMMARY OF COMMANDS

This appendix lists all EMACS commands alphabetically. Each command is followed by its corresponding function name and a brief description.

Note: The CONTROL key is represented by the caret (^) symbol in this appendix.

<u>Command</u>	<u>Function Name</u>	<u>Description</u>
^@	mark	Sets mark.
^A	begin_line	Moves point to beginning of line.
^B	back_char	Moves point back one character.
^C	reexecute	Reexecutes most recent command.
^D	delete_char	Deletes character after point.
^E	end_line	Moves point to end of line.
^F	forward_char	Moves point forward one character.
^G	abort_or_exit	Aborts most recent command or exits minibuffer.
{BS}	rubout_char	Deletes character before point.
{TAB}	type_tab	Moves point to next tab stop (LIBRARY).
^K	kill_line	Kills current line.
^L	refresh	Refreshes screen.
{CR}	cr	Inserts carriage return at point.
^N	next_line_command	Moves point to next line.
^O	open_line	Opens line without changing point.
^P		Breaks to PRIMOS command level.
^Q	^q_quote_command	Quotes next character (ignored on some terminals).
^R	reverse_search_command	Searches backward.
^S	^s_forward_search_command	Searches forward (stops output on some terminals).
^T	twiddle	Transposes two characters.
^U	multiplier	Multiplies count.
^V	next_page	Scrolls forward one page.
^W	kill_region	Moves region to kill ring.
^X^B	list_buffers	Lists buffers used in current editing session.
^X^C	quit	Leaves EMACS.
^X^E	primos_command	Executes a Primos command.

^X^F	find_file	Finds a file.
^X^G	ignore_prefix	Aborts the command.
^X^H	backward_kill_sentence	Kills a sentence backwards (LIBRARY).
^X-I	insert_tab	Inserts spaces to tab stop (LIBRARY).
^X^K	backward_kill_line	Kills from point to beginning of line (LIBRARY).
^X^L	lowercase_region	Converts region to lowercase (LIBRARY).
^X^O	delete_blank_lines	Deletes blank lines (LIBRARY)
^X^R	read_file	Reads a file into buffer.
^X^S	save_file	Saves a file.
^X^T	toggle_redisp	Toggles redisplay.
^X^U	uppercase_region	Converts region to uppercase (LIBRARY).
^X^V	view_file	Lets a user look at a file (LIBRARY).
^X^W	mod_write_file	Writes specified file (LIBRARY).
^X^X	exchange_mark	Exchanges mark and point.
^X^Z^A	backward_clause	Moves point backward one clause (LIBRARY).
^X^Z^E	forward_clause	Moves point forward one clause (LIBRARY).
^X^Z^F	get_filename	Inserts filename in buffer.
^X^Z^G	ignore_prefix	Aborts most recent command.
^X^Z^H	backward_kill_clause	Kills backward one clause at a time (LIBRARY).
^X^Z^K	forward_kill_clause	Kills forward one clause at a time (LIBRARY).
^X^Z^Y	yank_kill_text	Inserts text saved by view_kill_ring (LIBRARY).
^X^Z-<	mark_top	Marks top of buffer (LIBRARY)
^X^Z->	mark_bottom	Marks bottom of buffer (LIBRARY).
^X^Z-A	append_to_file	Appends current region to a file (LIBRARY).
^X^Z-F	take_right_margin	Sets right margin from current column (LIBRARY).
^X^Z-I	insert_buf	Inserts a buffer at point (LIBRARY).
^X^Z-K	view_kill_ring	Displays contents of kill ring (LIBRARY).
^X^Z-P	prepend_to_file	Prepends current region to a file (LIBRARY).
^X^Z-S	center_line	Centers current line (LIBRARY).
^X-(	collect_macro	Begins keyboard macro.
^X-)	finish_macro	Terminates keyboard macro.
^X-.	take_left_margin	Sets left margin from current column (LIBRARY).

<code>^X-1</code>	<code>mod_one_window</code>	Creates single window (LIBRARY).
<code>^X-2</code>	<code>mod_split_window</code>	Splits screen into two windows (LIBRARY).
<code>^X-3</code>	<code>mod_split_window_stay</code>	Splits screen into two windows and stays in current one (LIBRARY).
<code>^X-4</code>	<code>select_any_window</code>	Selects any window, cycles through all windows.
<code>^X-&lt;</code>	<code>horiz_left</code>	Shifts current window left (LIBRARY).
<code>^X-=</code>	<code>tell_position</code>	Displays line and character positions.
<code>^X-&gt;</code>	<code>horiz_right</code>	Shifts current window right (LIBRARY).
<code>^X-A</code>	<code>append_to_buf</code>	Appends current region to a buffer (LIBRARY).
<code>^X-B</code>	<code>select_buf</code>	Selects buffer.
<code>^X-D</code>	<code>explore</code>	Allows user to explore a directory (LIBRARY).
<code>^X-E</code>	<code>execute_macro</code>	Executes current keyboard macro.
<code>^X-F</code>	<code>set_right_margin</code>	Sets right margin to specified value (LIBRARY).
<code>^X-H</code>	<code>mark_whole</code>	Marks whole buffer.
<code>^X-I</code>	<code>insert_file</code>	Inserts file into buffer.
<code>^X-O</code>	<code>other_window</code>	Moves point to other window.
<code>^X-P</code>	<code>prepend_to_buf</code>	Prepends current region to a buffer (LIBRARY).
<code>^X-Q</code>	<code>quote_command</code>	Quotes next character.
<code>^X-R</code>	<code>repaint</code>	Moves cursor to specified line (LIBRARY).
<code>^X-S</code>	<code>save_file</code>	Saves file.
<code>^X-V</code>	<code>scroll_other_backward</code>	Scrolls other window backward (LIBRARY).
<code>^X-</code>	<code>backward_para</code>	Moves point backward one paragraph (LIBRARY).
<code>^X-]</code>	<code>forward_para</code>	Moves point forward one paragraph (LIBRARY).
<code>^Y</code>	<code>yank_region</code>	Yanks region from kill ring.
<code>^Z</code>	<code>prev_line_command</code>	Moves point up one line.
<code>ESC^D</code>	<code>kill_rest_of_buffer</code>	Kills from point to end of buffer (LIBRARY).
<code>ESC^G</code>	<code>ignore_prefix</code>	Aborts most recent command.
<code>ESC^O</code>	<code>split_line</code>	Breaks line at point and indents next (LIBRARY).
<code>ESC^V</code>	<code>scroll_other_forward</code>	Scrolls other window forward (LIBRARY).
<code>ESC^X-1</code>	<code>one_window</code>	Creates single window.
<code>ESC^X-2</code>	<code>split_window</code>	Splits screen into two windows.
<code>ESC^X-3</code>	<code>split_window_stay</code>	Splits screen into two windows and stays in current one.

ESC^Y	yank_minibuffer	Yanks minibuffer.
ESC ESC	pl_minibuffer	Enables PL minibuffer.
ESC {SP}	leave_one_white	Deletes all but one space around cursor (LIBRARY).
ESC %	query_replace	Executes query replace function.
ESC -	esc_minus	Prefix minus
ESC 0	esc_digit	Prefix digit
ESC 1	esc_digit	Prefix digit
ESC 2	esc_digit	Prefix digit
ESC 3	esc_digit	Prefix digit
ESC 4	esc_digit	Prefix digit
ESC 5	esc_digit	Prefix digit
ESC 6	esc_digit	Prefix digit
ESC 7	esc_digit	Prefix digit
ESC 8	esc_digit	Prefix digit
ESC 9	esc_digit	Prefix digit
ESC <	move_top	Moves to top of buffer.
ESC >	move_bottom	Moves to bottom of buffer.
ESC ?	explain_key	Explains specified key path.
ESC @	mark_end_of_word	Marks end of current word (LIBRARY).
ESC A	backward_sentence	Moves point back one sentence at a time (LIBRARY).
ESC B	back_word	Moves point back one word.
ESC C	capinitial	Capitalizes current word (LIBRARY).
ESC D	delete_word	Deletes a word.
ESC E	forward_sentence	Moves point forward one sentence (LIBRARY).
ESC F	forward_word	Moves point forward one word.
ESC G	goto_line	Moves point to specified line.
ESC H	mark_para	Marks a paragraph (LIBRARY).
ESC I	indent_relative	Indents current line relative to previous one.
ESC K	forward_kill_sentence	Kills a sentence forward from point (LIBRARY).
ESC L	lowercase_word	Converts word to lowercase (LIBRARY).
ESC M	back_to_nonwhite	Moves cursor to first non-blank character on line (LIBRARY).
ESC N	next_buf	Cycles forward through all buffers (LIBRARY).
ESC P	prev_buf	Cycles backward through all buffers (LIBRARY).
ESC Q	wrap_para	Fills a paragraph in fill mode (LIBRARY).
ESC R	reverse_search_command	Searches backward for specified string.
ESC S	forward_search_command	Searches forward for specified string.

ESC T	transpose_word	Transposes two words (LIBRARY).
ESC U	uppercase_word	Converts word to uppercase (LIBRARY).
ESC V	back_page	Scrolls backward one page.
ESC W	copy_region	Copies region to kill ring.
ESC X	extend_command	Extended command prefix.
ESC Y	yank_replace	Yanks previous region from kill ring and replaces previous yank.
ESC \	white_delete	Deletes space around point (LIBRARY).
ESC ^	merge_lines	Merges lines together (LIBRARY).
ESC ~	unmodify	Unmodifies buffer.
ESC {DEL}	rubout_word	Deletes word before cursor.
ESC ^	help_char	Accesses help function.
{DEL}	rubout_char	Deletes character before point.

#### Extended Commands

ESC X	write_file	Writes specified file.
ESC X	abort_command	Aborts command.
ESC X	primos_internal_screen	Executes internal PRIMOS command with terminal I/O.
ESC X	primos_external	Executes PRIMOS external command.
ESC X	primos_internal_quiet	Primos command without terminal I/O.
ESC X	primos_internal_como	Captures internal command with Comoutput.
ESC X	load_pl_source	Loads and compiles file of PL code.
ESC X	expand_macro	Expands keyboard macro into PEEL source code.
ESC X	delete_region	Deletes region without placing on kill ring.
ESC X	insert_version	Inserts version number of EMACS into buffer.
ESC X	apropos	Finds command descriptions.
ESC X	wallpaper	Inserts command descriptions into buffer.
ESC X	af	Evaluates active function.
ESC X	set_mode	Sets mode.
ESC X	set_mode_key	Sets permanent key for mode.
ESC X	set_key	Sets key in current buffer.
ESC X	set_permanent_key	Sets permanent key.
ESC X	popmark	Pops off of mark stack.
ESC X	setmark	Sets mark.
ESC X	pushmark	Sets mark pushes previous one off stack.
ESC X	vsplit	Splits screens vertically at point.

ESC X	replace	Replaces one string with another globally.
ESC X	date	Inserts date into buffer.
ESC X	dt	Inserts date and time into buffer.
ESC X	hcol	Sets or checks horizontal column.
FSC X	#	Tells if line numbering is in effect.
ESC X	#off	Turns off line numbering.
ESC X	#on	Turns on line numbering.
ESC X	2d	Tells if 2d is in effect.
ESC X	2doff	Turns off continuous lines.
ESC X	2don	Turns on continuous lines.
ESC X	dir	Lists directory.
ESC X	pl	Compiles PL procedure in buffer.
ESC X	tab	Moves point to next tab stop.
ESC X	load	Loads library files.
ESC X	untidy	Unjustifies a paragraph.
ESC X	case_on	Enables case matching during search function.
ESC X	case_off	Disables catch matching during search function.
ESC X	case?	Tells if case matching occurs during search function.
ESC X	all_modes_off	Turns off all modes (LIBRARY).
ESC X	back_tab	Moves point to previous tab stop (LIBRARY).
ESC X	default_tabs	Sets up default tabs every five spaces (LIBRARY).
ESC X	describe	Displays information about functions and commands (LIBRARY).
ESC X	dump_file	Converts to fasdump file (LIBRARY).
ESC X	europe_dt	Inserts date in European format (LIBRARY).
ESC X	fill_on	Turns on fill mode in current buffer.
ESC X	get_tab	Restores named tabs (LIBRARY).
ESC X	hscroll	Sets horizontal column at point (LIBRARY).
ESC X	lisp_on	Turns on Lisp mode (LIBRARY).
ESC X	load_compiled	Loads a fasdump file (LIBRARY).
ESC X	new_features	Lists new EMACS features (LIBRARY).
ESC X	overlay_on	Turns on overlay mode (LIBRARY).

ESC X	reset	Resets windows and columns (LIBRARY).
ESC X	save_all_files	Saves all files, ignores modified buffers (LIBRARY).
ESC X	save_tab	Saves tab stops for future use (LIBRARY).
ESC X	set_hscroll	Lets you specify hcol position (LIBRARY).
ESC X	setft	Sets tabs from a line (LIBRARY).
ESC X	settab	Sets tabs to specified values (LIBRARY).
ESC X	settabs_from_table	Sets tabs from a line (LIBRARY).
ESC X	set_left_margin	Sets left margin (LIBRARY).
ESC X	sort_dt	Puts date into sort order (LIBRARY).
ESC X	tablist	Sets tabs from a numbered list (LIBRARY).
ESC X	tell_modes	Tells which modes are on (LIBRARY).
ESC X	tell_left_margin	Displays column number of left margin (LIBRARY).
ESC X	tell_right_margin	displays column number of right margin (LIBRARY).
ESC X	trim_dt	Inserts date without hours (LIBRARY).
ESC X	trim_date	Inserts date without day (LIBRARY).

## APPENDIX C

## COMPLETE CROSS-REFERENCE LIST OF COMMANDS

Aborting Commands

CNTRL-G	Aborts command; exits minibuffer
CNTRL-P	Aborts command; goes back to PRIMOS command level

Basic Cursor Movement

CNTRL-A	Moves point to beginning of line
CNTRL-B	Moves point backward one character
CNTRL-E	Moves point to end of line
CNTRL-F	Moves point forward one character
CNTRL-N	Moves point to next line
CNTRL-Z	Moves point up one line
ESC <	Moves point to beginning of buffer
ESC >	Moves point to end of buffer

Buffers

CNTRL-X A	Appends current region to buffer
CNTRL-X B	Selects or creates buffer
CNTRL-X CNTRL-B	Lists buffers
CNTRL-X CNTRL-Z I	Inserts buffer at point
CNTRL-X P	Prepends current region to buffer
ESC CNTRL-D	Kills from point to end of buffer
ESC N	Cycles through all buffers beginning with next one
ESC P	Cycles through all buffers beginning with previous one
ESC ~	Unmodifies buffer



Case Conversion

CNTRL-X CNTRL-L	Converts region to lowercase
CNTRL-X CNTRL-U	Converts region to uppercase
ESC C	Capitalizes first letter of word
ESC L	Changes word to lowercase
ESC U	Changes word to uppercase

Executing Primos Commands

CNTRL-X CNTRL-E	Executes PRIMOS command
-----------------	-------------------------

Exiting

CNTRL-X CNTRL-C	Leaves EMACS
-----------------	--------------

Files

CNTRL-X CNTRL-F	Finds file and places it in current buffer
CNTRL-X CNTRL-R	Reads file into current buffer
CNTRL-X CNTRL-S	Saves file under current pathname
CNTRL-X CNTRL-W	Writes file under pathname specified by user
CNTRL-X I	Inserts file at point
CNTRL-Z CNTRL-Z A	Appends current region to a file
ESC X Save_all_files	Saves all modified files

Formatting

CNTRL-T	Transposes two characters preceding point
CNTRL-X CNTRL-Z S	Centers line according to current fill prefix
ESC I	Indents line relative to one above it
ESC Q	Fills paragraph according to current fill prefix

ESC X Untidy	Unjustifies a paragraph
ESC T	Transposes two words between point
<u>Help</u>	
CNTRL-__	Accesses help options
ESC ?	Describes character command
ESC X Apropos	Lists commands relating to specified subject
ESC X Describe	Describes functions relating to specified subject
ESC X New_features	Lists new features in EMACS
ESC X Wallpaper	Inserts list of all EMACS commands at point
<u>Information</u>	
CNTRL-X =	Gives line and character position of point
CNTRL-X CNTRL-B	Lists all buffers used in current editing session
ESC X Tell_modes	Tells which modes are in effect
ESC X Wallpaper	Inserts list of all EMACS commands at point
ESC X Date	Inserts date into buffer
ESC X Dir	Inserts full pathname of specified file into buffer
ESC X Dt	Inserts date and time into buffer
ESC X Europe_date	Inserts date into buffer
ESC X Insert_version	Inserts current version of EMACS into buffer
ESC X Sort_dt	Inserts date into buffer
ESC X Trim_date	Inserts date into buffer
ESC X Trim_dt	Inserts date into buffer

Inserting Lines and Text

CTRL-O	Inserts new line
CTRL-X A	Appends current region to buffer
CTRL-X CTRL-Z A	Appends current region to a file
CTRL-X CTRL-Z I	Inserts a buffer at point
CTRL-X CTRL-Z P	Prepends current region to a file
CTRL-X I	Inserts a file at point
CTRL-X P	Prepends current region to buffer
{RETURN}	Carriage return; inserts new line

Keyboard Macros

CTRL-X (	Begins keyboard macro
CTRL-X )	Ends keyboard macro
CTRL-X E	Executes most recent keyboard macro
ESC X Load_pl_source	Compiles source code in specified file
ESC X Expand_macro	Expands most recent macro into PEEL source code
ESC X PL	Compiles source code in current buffer
ESC X Set_permanent_key	Binds macro to keypath you specify
ESC X Set_key	Binds macro to keypath for current buffer only

Killing and Deleting

CTRL-D	Deletes character after point
CTRL-H	Deletes character before point
CTRL-K	Kills line
CTRL-W	Moves region to kill ring
CTRL-X CTRL-K	Kills from point back to beginning of line

CTRL-X CTRL-O	Deletes blank lines
CTRL-X CTRL-Z K	Shows contents of kill ring
ESC CTRL-D	Kills from point to end of buffer
ESC X Delete_region	Deletes region
ESC \	Deletes whitespace around cursor
{BACKSPACE}	Deletes character before point
{DEL}	Deletes character before point
{RUBOUT}	Deletes character before point
 <u>The Mark and the Region</u>	
CTRL-@	Sets mark
CTRL-U CTRL-@	Moves point to previous mark
CTRL-W	Moves region to kill ring
CTRL-X A	Appends current region to buffer
CTRL-X CTRL-L	Converts region to lowercase
CTRL-X CTRL-U	Converts region to uppercase
CTRL-X CTRL-X	Exchanges mark and point
CTRL-X CTRL-Z <	Marks top of buffer
CTRL-X CTRL-Z >	Marks bottom of buffer
CTRL-X CTRL-Z A	Appends current region to file
CTRL-X CTRL-Z CTRL-Y	Inserts text saved by view_kill_ring
CTRL-X CTRL-Z K	Displays contents of kill ring
CTRL-X CTRL-Z P	Prepends current region to file
CTRL-X H	Marks whole buffer
CTRL-X P	Prepends current region to buffer
CTRL-Y	Yanks region from kill ring
ESC @	Marks end of current word
ESC H	Marks paragraph

ESC W	Copies region to kill ring
ESC X Delete_region	Deletes region
ESC X Popmark	Pushes mark off mark ring
ESC X Pushmark	Pushes mark on mark ring
ESC X Setmark	Sets mark
ESC Y	Replaces kill ring entry with earlier kill

#### Modes and Mode-Like Environments

ESC X Set_mode	Puts specified mode in effect
ESC X Set_mode_key	Binds function to keypath in specified mode only
ESC X All_modes_off	Turns off all modes

#### Overlay Mode

ESC X Overlay_on	Turns on overlay mode
ESC X Overlay_off	Turns off overlay mode

#### Fill Mode

ESC X Fill_on	Turns on fill mode
ESC X Fill_off	Turns off fill mode
CNTRL-.	Sets left fill margin to column you specify
CNTRL-X CNTRL-Z F	Sets right fill margin to column you specify
ESC Q	Fills paragraph according to fill values
ESC CNTRL-I	Indents to fill prefix

Explore Mode

ESC X Explore Turns on explore mode

Line Numbering

ESC X #On Turns on line numbering

ESC X #Off Turns off line numbering

ESC X # Tells if line numbering is in effect

ESC G Goes to line number specified by argument

Continuous Lines

ESC X 2don Turns on continuous lines

ESC X 2doff Turns off continuous lines

ESC X 2d Tells if continuous lines are in effect

Moving Text

CNTRL-W Moves region to kill ring

CNTRL-Y Yanks region from kill ring

ESC W Copies region to kill ring

ESC Y Replaces kill ring entry with earlier kill

Prefix Characters

CNTRL-U

CNTRL-X

CNTRL-X CNTRL-Z

ESC

ESC X

Reexecuting Commands

CNTRL-C Reexecutes last command typed

Replacing

ESC X Replace	Replaces <string1> with <string2> in region
ESC %	Searches for <string1> and replaces with <string2> on request

Saving

CNTRL-X CNTRL-S	Saves file
CNTRL-X CNTRL-W	Writes file
ESC X Save_all_files	Saves all modified files; ignores modified buffers

Scrolling and Display Control

CNTRL-L	Refreshes screen
CNTRL-V	Advances to next screen
CNTRL-X R	Repaints screen
CNTRL-X V	Scrolls other window backward
ESC CNTRL-V	Scrolls other window forward
ESC V	Moves backward one page
ESC X Reset	Resets horizontal column to 1
ESC X Set_hscroll	Sets horizontal column to value you specify

Searching

CNTRL-R	Searches backward through buffer for <string>
CNTRL-S	Searches forward through buffer for <string>
ESC %	Searches for <string1> and replaces with <string2> on request
ESC R	Searches backward through buffer for <string>
ESC S	Same as CNTRL-S

ESC X Replace Searches for <string1> and replaces it with <string2> globally

### Slow Terminals

CNTRL-X CNTRL-T Suppresses display until typed again

CNTRL-X CNTRL-Z CNTRL-V Shows current state of display

### Tabs

CNTRL-I Moves point right to nearest tab stop

CNTRL-X CNTRL-I Indents to nearest tab stop

ESC X Back\_tab Moves left to nearest tab stop

ESC X Default\_tabs Resets tabs to default value

ESC X Get\_tab Retrieves stored tab stops from a file

ESC X Save\_tab Saves current tab stops in a file

ESC X Settab Sets tab stops to values you specify

ESC X Settabs\_from\_table Sets tab stops from current line

ESC X Tablist Sets tab stops from list of column numbers

{TAB} Moves point right to nearest tab stop

### Whitespace and Indentation

CNTRL-X CNTRL-I Indents to nearest tab stop

CNTRL-X CNTRL-O Deletes blank lines

ESC CNTRL-I Indents to fill prefix

ESC I Indents current line relative to line above it

ESC \ Deletes whitespace around point

ESC ^ Merges two lines together



Windows

CTRL-X 1	Makes one window again
CTRL-X 2	Splits screen into 2 windows
CTRL-X 3	Splits screen into 3 windows; point stays in window 1
CTRL-X 4	Cycles through all windows
CTRL-X O	Moves point to other window
CTRL-X V	Scrolls other window backward
ESC CTRL-V	Scrolls other window forward
ESC X Vsplit	Splits screen vertically at point

Words, Sentences, Clauses, and Paragraphs

CTRL-E	Moves point forward one clause
CTRL-X CTRL-Z CTRL-A	Moves point backward one clause
CTRL-X CTRL-Z CTRL-H	Kills backward one clause
CTRL-X CTRL-Z CTRL-K	Kills forward one clause
CTRL-X [	Moves point forward one paragraph
CTRL-X ]	Moves point backward one paragraph
CTRL-X {BACKSPACE}	Kills backward one sentence
ESC A	Moves point backward one sentence
ESC B	Moves backward one word
ESC D	Deletes word after point
ESC E	Moves forward one sentence
ESC F	Moves forward one word
ESC H	Marks paragraph
ESC K	Kills forward one sentence
ESC Q	In fill mode, fills a paragraph
ESC X Untidy	Unjustifies a paragraph

{BACKSPACE}	Deletes word preceding point
{DEL}	Deletes word preceding point
{RUBOUT}	Deletes word preceding point

## INDEX

- Aborting commands 6-1, 6-2
- Adding blank lines 4-14, 4-15
- Adding text to a file 4-3,  
4-4, 4-12 to 4-15
- Apropos prompt 6-5
- Arguments, how to specify 3-9
- BACKSPACE key 2-4, 3-3
- Backward deletion, by line 5-10
- Backward movement:
  - by character 3-6 to 3-8
  - by paragraph 9-3, 9-4
  - by screen 4-7
  - by searching 4-11
  - by sentence 9-1
  - by word 3-10 to 3-13
  - to top of file 4-5
- Beginning a session 2-4
- Blank lines, how to insert  
4-12, 4-14, 4-15
- Blank space, how to type 2-3
- Blocks or text, how to delete  
5-12 to 5-16, 5-20, 5-21
- Blocks of text, how to move  
5-12, 5-13
- Bottom of file, moving to 4-4
- Buffer, definition of 11-1
- Buffers:
  - how to create 11-1 to 11-3
  - how to name 11-2 to 11-3
  - use of 11-1 to 11-11
- Case conversion commands 9-10  
to 9-15
- Centering lines 8-3, 8-4
- Character movement:
  - backward 3-6
  - forward 3-14
- Combinations of commands:
  - how to type 2-3
- Commands:
  - how specified 2-1, 2-2
- CONTROL commands:
  - comparison with ESCAPE commands  
3-19
  - examples of 2-2, 2-3
  - how to type 2-2, 2-3
- CONTROL key, location of 2-1
- CONTROL-@ command 5-13, 5-16,  
5-21, 7-1 to 7-7
- CONTROL-A command 4-18, 4-19
- CONTROL-B command 3-6 to 3-8,  
3-24
- CONTROL-C command 4-10, 4-19
- CONTROL-D command 3-25 to  
3-28, 3-50, 4-17
- CONTROL-E command 4-18, 4-19
- CONTROL-F command 3-13 to  
3-15, 3-24
- CONTROL-G command 6-2, 6-9
- CONTROL-I command 8-5 to 8-15,  
8-21
- CONTROL-K command 5-1, 5-2,  
5-6 to 5-8, 5-21
- CONTROL-L command 3-49, 6-2,  
6-8
- CONTROL-N command 3-23 to  
3-24, 3-50
- CONTROL-O command 4-12, 4-14,  
4-15, 4-19
- CONTROL-P command 3-48, 3-49,  
6-1, 6-9

INDEX

CONTROL-R command	4-11, 4-19	CONTROL-X CONTROL-Z S command	8-3, 8-4, 8-21
CONTROL-S command	4-8, 4-9, 4-19	CONTROL-X E command	12-5 to 12-6, 12-15
CONTROL-T command	9-8, 9-9, 9-15	CONTROL-X O command	11-16, 11-19
CONTROL-U command	9-10 to 9-13	CONTROL-X V command	11-16, 11-19
CONTROL-V command	4-6, 4-19	CONTROL-X [ command	9-3 to 9-4, 9-15
CONTROL-W command	5-16, 5-20, 5-21	CONTROL-X ] command	9-4 to 9-5, 9-15
CONTROL-X 0 command	11-14	CONTROL-Y command	5-3, 5-9, 5-11, 5-12, 5-15, 5-17 to 5-19, 5-21
CONTROL-X 1 command	11-19	CONTROL-Z command	3-20, 3-21, 3-24, 3-50
CONTROL-X 2 command	11-12, 11-19	CONTROL-_ command	6-8, 6-9
CONTROL-X B command	11-2 to 11-6, 11-19	CONTROL-_A command	6-5, 6-6, 6-9
CONTROL-X CONTROL-B command	11-7	CONTROL-_C command	6-3, 6-4, 6-8, 6-9
CONTROL-X CONTROL-C command	3-48, 3-50	CONTROL-_L command	6-6, 6-9
CONTROL-X CONTROL-E command	10-6, 10-7	Conventions:	
CONTROL-X CONTROL-F command	3-3, 3-50, 4-1, 4-2, 4-19, 10-6, 11-19	command	2-1, 2-2
CONTROL-X CONTROL-I command	8-14, 8-15, 8-21	typing	2-2
CONTROL-X CONTROL-R command	11-2, 11-19	Converting words to lowercase	9-13 to 9-15
CONTROL-X CONTROL-S command	3-47, 3-48, 3-50, 11-19	Converting words to uppercase	9-10 to 9-13
CONTROL-X CONTROL-W command	11-9 to 11-10, 11-19	Correcting mistakes	2-4, 3-25, 9-5 to 9-9
CONTROL-X CONTROL-X command	5-16, 5-21, 7-1 to 7-7	Creating a file	3-3
		Cursor	3-2, 3-4, 3-6

## INDEX

- Cursor movement:
  - backward 3-6 to 3-12, 4-18
  - forward 3-13 to 3-17, 4-4, 4-5, 4-19
  - vertical 3-20 to 3-23
- Default tab stops 8-4 to 8-10
- DEL key, use of 3-29
- Deleting blocks of text 5-13 to 5-14, 5-20
- Deleting characters 3-25, 3-27 to 3-29
- Deleting lines 5-1, 5-2, 5-6 to 5-8, 5-10
- Deleting words 3-25, 3-31 to 3-38
- Display control 4-6, 4-7, 10-1 to 10-3, 11-16, 11-17
- Downward movement 3-22, 3-23
- Editing an existing file 4-1
- Editing session, definition of 2-4
- EMACS session, definition of 2-4
- End of line, how to type 2-3
- Ending an editing session 2-4, 3-48
- Entering text 3-3, 3-4, 3-30, 3-39 to 3-45, 3-47
- Errors, how to correct 2-4, 3-25
- ESCAPE X Fill\_on command 8-20
- ESCAPE % command 7-3 to 7-9
- ESCAPE 0 CONTROL-K command 5-10
- ESCAPE < command 4-5, 4-6, 4-19
- ESCAPE <digit> command 3-8, 3-12, 3-14, 3-18, 3-24, 3-50
- ESCAPE > command 4-4, 4-5, 4-19
- ESCAPE A command 9-1, 9-15
- ESCAPE B command 3-10, 3-24, 3-50
- ESCAPE commands:
  - examples of 2-2
  - how to type 2-2
- ESCAPE CONTROL-V command 11-17, 11-19
- ESCAPE D command 3-31 to 3-35, 3-50
- ESCAPE E command 9-2, 9-3, 9-15
- ESCAPE F command 3-14, 3-15, 3-24, 3-50
- ESCAPE I command 8-1, 8-2, 8-21
- ESCAPE key:
  - location of 2-1
- ESCAPE L command 9-13 to 9-15
- ESCAPE Q command 8-16 to 8-20
- ESCAPE S command 4-8
- ESCAPE T command 9-8, 9-15
- ESCAPE U command 9-10, 9-15
- ESCAPE V command 4-7
- ESCAPE X Fill\_off command 8-20, 8-21
- ESCAPE X Fill\_on command 8-17, 8-21

## INDEX

- ESCAPE X Load command 3-3,  
4-3, 6-8
- ESCAPE X Overlay\_off command  
10-4, 10-7
- ESCAPE X Overlay\_on command  
10-4, 10-7
- ESCAPE X Replace command 7-2,  
7-3, 7-7
- ESCAPE X Reset command 10-3,  
10-7
- ESCAPE X Setft command 8-21
- ESCAPE X Settab command 8-4 to  
8-11, 8-21
- ESCAPE X Set\_hscroll command  
10-1 to 10-3, 10-7
- ESCAPE {BACKSPACE} command 3-50
- Executing PRIMOS commands 10-5  
to 10-6
- Exiting EMACS 3-48, 3-49
- Explain key prompt 6-3, 6-4
- File written prompt 3-47
- Fill mode 8-16 to 8-20
- Filling paragraphs 8-16 to  
8-20
- Find file prompt 3-1, 3-3
- Finding a file 4-1, 4-2
- Formatting commands 8-1 to  
8-21
- Forward movement:
  - by paragraph 9-4 to 9-5
  - by screen 4-6
  - by searching 4-9
  - by sentence 9-2 to 9-3
  - by word 3-16, 3-17
- Forward search prompt 4-8
- Getting back lost text 5-11,  
5-12
- Indenting lines 8-1, 8-2
- Inserting blank lines 4-12,  
4-14
- Inserting text 3-30, 3-47,  
4-12 to 4-14
- Inserting text at the beginning  
of a file 4-15
- Invisible line separator 4-18,  
5-1, 5-2
- Invisible mark 5-13
- Keyboard macros, use of 12-1  
to 12-15
- Killing lines 5-1, 5-2, 5-10
- Line deletion 5-1, 5-2
- Line movement 3-20 to 3-23
- Listing buffers 11-7
- LOAD command 4-3
- Loading special libraries 4-3
- Lost text, how to retrieve 5-12
- Lowercase conversion 9-13 to  
9-15
- Macros, keyboard 12-1 to 12-15
- Mark 5-13, 5-16
- Mistakes, how to correct 2-4,  
3-25
- Mode line 3-2, 11-1
- Moving text 5-12, 5-13, 5-18

## INDEX

- Moving text to tab stops 8-14, 8-15
- Multiple windows, how to use 11-12 to 11-19
- Naming a buffer 11-2 to 11-6
- Naming a File 3-3
- Overlay mode 10-3 to 10-5
- PL prompt, how to remove 6-2
- PL:
  - how to erase 2-2
- Point 5-14, 5-16
- Primos command prompt 10-6
- PRIMOS commands, how to execute while in EMACS 10-5 to 10-6
- PT45 keyboard 2-1
- Query replace command 7-7 to 7-9
- Reexecuting a command 4-10
- Refreshing the screen 6-2
- Regions of text 5-12 to 5-20
- Reinserting text 5-3, 5-4, 5-9, 5-10, 5-11, 5-15, 5-17 to 5-19
- Repeating a command 3-8, 3-13, 3-14
- Replacement commands 7-1 to 7-7
- Replacement, global 7-2, 7-3
- Replacement, query 7-4 to 7-9
- Restarting EMACS 4-1 to 4-3
- RETURN key, use of 2-3, 3-4
- Reverse search 4-11
- Reverse search prompt 4-11
- Saving a document 3-47
- Saving a file 3-47
- Saving text in a buffer 11-9 to 11-10
- Scrolling the screen:
  - backward 4-7, 11-16
  - forward 4-6, 11-17
  - horizontally 10-1 to 10-3
- SPACE key, use of 2-3
- Split screen, how to use 11-12 to 11-19
- Start-up commands 3-1
- Starting EMACS 3-1
- Switching mark and point 5-16
- Tabs:
  - command options 8-7
  - default settings 8-5 to 8-10
  - how to set 8-11 to 8-12
  - how to set from table 8-12 to 8-14
  - use of 8-4 to 8-15
- Terminal session, definition of 2-4
- Terminal type, how to specify 3-1
- Top of file, moving to 4-5
- Transposing characters 9-5 to 9-7
- Transposing words 9-7 to 9-9
- Transpositions, how to fix 9-5 to 9-9
- TTP command line option 3-1, 4-1

## INDEX

Typing conventions 2-2, 2-3

Typing mistakes, how to correct  
2-4, 3-25

Uppercase conversion 9-10

Vertical movement:  
down 3-22, 3-23  
up 3-20, 3-21

Windows, use of 11-12 to 11-19

Word deletion:  
backward 3-37, 3-38  
forward 3-25, 3-31 to 3-36

Yanking text 5-9, 5-10

{BACKSPACE} command 3-28,  
3-29, 3-50

{RETURN} command 4-12

{RETURN}, when to type 2-3

{TAB} command 8-4 to 8-15